Facultade de Informática
UNIVERSIDADE DA CORUÑA

TRABALLO FIN DE GRAO
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN ENXEÑARÍA DE COMPUTADORES

Monitorización y control de explotación de ganadería extensiva

Estudante: Manuel Couto Carril
Dirección: Carlos J. Escudero Cascón

A Coruña, febreiro de 2021.

Dedicatoria

A Tania, por creer en mí.

A Nisi, por su apoyo incondicional.

Agradecimientos

Quisiera agradecer a Carlos J. Escudero, director de este TFG, por abrirme las puertas de este fascinante mundo. Por inspirarme. Por contagiarme su pasión por su trabajo. Por ayudarme los domingos. Por confiar en mí.

A Álvaro, por plantar la semilla de este trabajo. Gracias por cederme las instalaciones y acompañarme en este camino.

Resumen

Actualmente existe una gran cantidad de soluciones tecnológicas que permiten la implantación de sistemas Internet of Things (IoT) en casas, edificios y ciudades donde se supone una alta calidad en las comunicaciones y, en ocasiones, con posibilidad de conexión a la red eléctrica. El abanico de posibilidades se reduce especialmente cuando las comunicaciones son en exteriores, a largas distancias, en lugares sin cobertura móvil y los dispositivos deben ser alimentados con pequeñas baterías.

La ejecución de este proyecto consiste en el diseño e implementación de un sistema de monitorización y control de una explotación de ganadería extensiva que permite a los ganaderos vigilar la dispersión del ganado, monitorizar los tanques de agua, controlar las vallas electrificadas y llevar un registro de las zonas de pastoreo; posibilitando así un mayor control sobre los animales y la explotación en general.

La explotación ganadera, para la que está diseñado este sistema, está desplegada en una extensa zona del interior de Galicia, que intercala distintas fincas de pasto con zonas boscosas. Este entorno de aplicación supone un reto para las Tecnologías de la información y la comunicación (TIC) ya que el sistema debe ser desasistido, alejado de puntos urbanos, con existencia de terrenos montañosos, en ocasiones cubiertos de vegetación, que dificultan las comunicaciones y, en general, con usuarios poco expertos en el manejo de nuevas tecnologías.

Para llevar el proyecto a cabo, se considera hardware de bajo consumo que permite el uso de dispositivos pequeños, con baterías que proporcionan gran autonomía y transmisiones que cubran un área muy extensa.

Abstract

Currently there is a large number of technological solutions that allow the implementation of Internet of Things (IoT) systems in homes, buildings and cities where high quality communications are assumed and, on occasions, with the possibility of connection to the electricity grid. The range of possibilities is reduced especially when communications are outdoors, over long distances, in places without mobile coverage and the devices must be powered by small batteries.

The execution of this project consists of the design and implementation of a monitoring and control system of an extensive livestock farm that allows farmers to monitor the dispersal of livestock, monitor water tanks, control electrified fences and keep a record of grazing areas; thus enabling greater control over the animals and the farm in general.

The livestock farm, for which this system is designed, is deployed in a large area in the interior of Galicia, which intersperses different pasture farms with wooded areas. This application environment represents a challenge for Information and Communications Technology (ICT) since the system must be unassisted, away from urban points, with the existence of mountainous terrain, sometimes covered with vegetation, which makes communication difficult and users who are not very expert in handling of new technologies.

To carry out the project, low consumption hardware is considered that allows the use of small devices, with batteries that provide great autonomy and transmissions that cover a very large area.

Palabras clave:

- ganadería
- extensiva
- monitorización
- control
- geolocalización
- lorawan
- IoT

Keywords:

- livestock
- extensive
- monitoring
- control
- geolocation
- lorawan
- IoT

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

El control y monitorización de explotaciones de ganadería extensiva puede suponer una serie de ventajas considerable tanto para el confort de los animales como para la comodidad y economía de los ganaderos, optimizando los recursos de la explotación.

En el presente proyecto se tratará el proceso de análisis, diseño e implementación de un sistema desasistido que permita monitorizar y controlar diversos aspectos fundamentales de este tipo de empresas.

1.1 Origen del proyecto

La ganadería presente en Galicia es, tradicionalmente, intensiva [1], en la cual el ganado se halla estabulado, bajo unas condiciones creadas de forma artificial. Muchas de estas ganaderías gozan de grandes avances tecnológicos para el cuidado de las reses, como redes de sensores y actuadores, que facilitan la gestión de las explotaciones, ayudando así a maximizar su productividad.

La creciente despoblación en el rural gallego y el consiguiente abandono de terrenos de antiguos ganaderos provoca que grandes extensiones de campo queden sin explotar. Esta situación unida a un clima poco contrastado en la mayor parte de Galicia [2] y el aumento mundial de la demanda de productos ecológicos u obtenidos mediante medios de producción sostenibles, propicia que emerjan otras formas de ganadería, como puede ser la ganadería extensiva. En este tipo de ganadería las reses gozan de semilibertad, en grandes extensiones de terreno para pastoreo, siendo ésta su principal forma de alimentación. En éstas, el área de desarrollo tecnológico está menos desarrollada, siendo gestionadas, la mayor parte de ellas, de forma totalmente manual.

En la figura 1.1 se muestra un extracto del "Plan Xeral de Ordenación Municipal de Santa Comba" [3], en el cual se ve la distribución de las diferentes zonas de calificación territorial pertenecientes al ayuntamiento de Santa Comba, uno de los numerosos municipios gallegos dedicado, tradicionalmente, a la ganadería y la agricultura; con una distribución del suelo similar a la del resto de pequeños núcleos de Galicia. Las calificaciones consideradas más relevantes para este caso de estudio son: suelo urbano, suelo rústico, suelo de protección agropecuaria y el suelo de protección forestal; todas claramente diferenciadas y separadas.

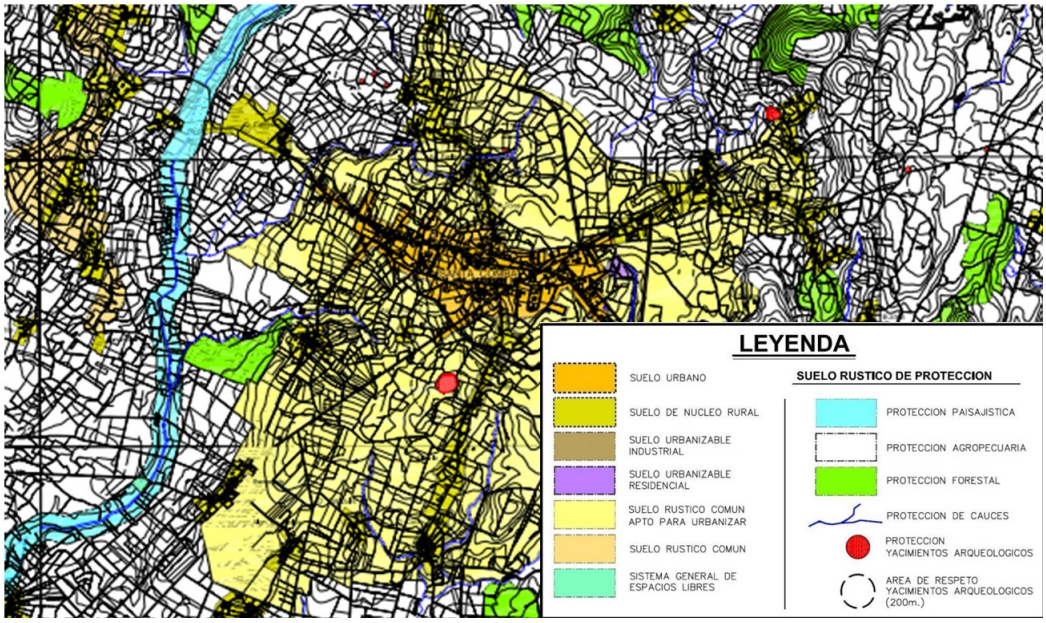


Figura 1.1: Extracto del "Plan Xeral de Ordenación Municipal de Santa Comba".

Según recoge la Ley 2/2016 [4], las explotaciones ganaderas de nueva construcción, pueden ser llevadas a cabo (con diferentes niveles de restricción) en cualquiera de los suelos anteriormente mencionados, a excepción del urbano, del cual, además, es obligatorio dejar una distancia mínima para nuevas construcciones, que varía según el municipio. Este hecho, obliga a las nuevas explotaciones, a establecerse en áreas alejadas de puntos urbanos y, en muchas ocasiones, con grandes dificultades para conseguir el suministro de energía eléctrica y el abastecimiento de agua. A estas dificultades, hay que sumarle, que en las explotaciones de ganadería extensiva, los animales pueden tener disponibles decenas o cientos de hectáreas en las que vivir, lo que imposibilita la viabilidad económica de la extensión del suministro eléctrico y el abastecimiento de agua a toda la explotación.

1.2 Motivación

Para los ganaderos dedicados a las explotaciones extensivas, tener controlado el ganado y diversos aspectos de la explotación de forma remota cambiará totalmente el enfoque diario de trabajo, consiguiendo: evitar la pérdida de animales; posibilitar la detección de enfermedades de forma precoz; monitorizar parámetros de la explotación; detectar fallos en las instalaciones. También se optimizarán los desplazamientos realizados para la ejecución de tareas que permitan ser automatizadas o ejecutadas de forma remota.

La migración de tareas manuales hacia las TIC conllevará el ahorro de mano de obra, lo que reducirá considerablemente los costes de producción de la empresa e incrementará su competitividad en el mercado. Además, el ganadero podrá invertir el tiempo ahorrado gracias a la tecnología, en realizar trabajo productivo para la empresa.

1.3 Estado del arte

Para la implantación de las TIC en las explotaciones de ganadería extensiva existen pocas soluciones en el mercado. En España, estos tipos de ganadería, generalmente, sufren un gran atraso tecnológico con respecto a las de ganadería intensiva.

Unas de las empresas con un producto diseñado para cubrir las necesidades de ganaderías extensivas son la empresa madrileña Digitanimal S.L. [5] y la empresa catalana Accent Advanced Systems S.L. [6].

Estas empresas proveen a los ganaderos de dispositivos, resistentes a golpes y a condiciones meteorológicas adversas, ubicados en collares para colocar al ganado, diseñados para transmisiones a largas distancias y baterías de larga duración. Disponen de aplicaciones en las que, entre otras funcionalidades, permiten al ganadero geolocalizar a cada animal, crear vallas virtuales y recibir notificaciones en caso de que alguna res se encuentre fuera de la parcela indicada. La aplicaciones suelen facilitar al ganadero el historial de los últimos recorridos de los animales y análisis de datos.

Se detecta en el análisis de estos productos, que son soluciones con poca flexibilidad, abordando aspectos como la localización del ganado y el análisis de distintos parámetros de los animales, sin tener en cuenta la monitorización y el control de la explotación. Además, el coste que debe asumir una explotación tanto para la implantación de este sistema [7], como para su mantenimiento mensual [8] es muy elevado.

1.4 Objetivos

A continuación, se enumeran los objetivos principales de este proyecto:

- Analizar el problema y decidir las tecnologías que se ajustan mejor.
- Aprender y familiarizarse con las tecnologías seleccionadas.
- Diseñar la arquitectura del sistema.
- Diseñar y construir un sistema para monitorizar el nivel de llenado y temperatura de un depósito de agua móvil, notificando al ganadero si el nivel de agua del mismo es inferior a una cantidad predefinida.
- Diseñar y construir un sistema que permita controlar la valla eléctrica de forma remota.
- Utilizar dispositivos de geolocalización y transmisión.
- Integrar todos los dispositivos hardware.
- Poner en marcha el sistema y sus funcionalidades: plataforma de IoT, alertas, panel de control, etc.

1.5 Metodología de desarrollo del proyecto

El desarrollo del proyecto se realizó con el director del mismo usando metodologías ágiles para la toma de decisiones y evolución, utilizando un sistema de desarrollo adaptativo. Este modelo de desarrollo fue adoptado debido al desconocimiento de la forma exacta de crear el sistema final en sus inicios. La metodología está formada por las fases que, a continuación, se indican:

- Análisis del estado del arte y recogida de requisitos.
- Diseño del sistema hardware-software.
- Pruebas de concepto y selección de tecnologías.
- Construcción y desarrollo de los prototipos.
- Pruebas de unidad.
- Integración de funcionalidades a la aplicación servidora.

- Pruebas de integración, análisis general y exposición de conclusiones.
- Documentación.

Para la planificación del proyecto se utiliza el diagrama de Gantt, representado en la figura 1.2, donde cada parte del desarrollo del proyecto aparece reflejada como un paquete de trabajo. Está compuesto por un eje vertical donde se establecen los paquetes de actividades que forman el proyecto y un eje horizontal, dividido en semanas, que muestra la duración de cada paquete y del proyecto en general.

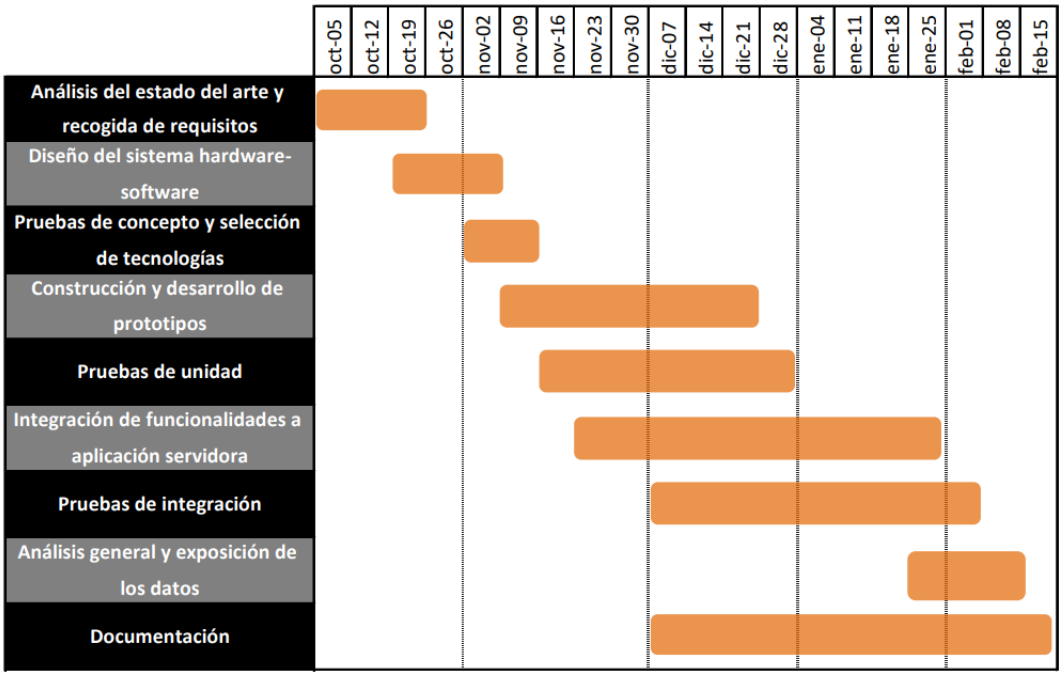


Figura 1.2: Diagrama de Gantt. El eje vertical contiene los paquetes de tareas a realizar y el horizontal las semanas disponibles para el desarrollo del proyecto.

1.6 Herramienta

Este proyecto consiste en el diseño y desarrollo de un sistema de monitorización y control de una explotación de ganadería extensiva.

El "backend" de la aplicación está centralizado en un pequeño servidor ubicado en las oficinas centrales de la empresa ganadera. La lógica de la aplicación servidora corre sobre una plataforma de IoT, que también proporciona una aplicación cliente para teléfonos inteligentes y tabletas. Esta aplicación dispone de una interfaz amigable, que el ganadero puede manejar

desde cualquier lugar con conexión a internet, a través de cualquier navegador web o a través de la aplicación disponible para Android e iOS.

El sistema trata los aspectos que se detallan a continuación:

- Identificación y geolocalización de reses. Funcionalidad que permite al ganadero saber dónde se encuentran los animales en cada momento y monitorizar sus recorridos diarios.

- Monitorización de temperatura y nivel de agua de los tanques. El ganadero tiene monitorizado en tiempo real el nivel de agua de cada tanque y su temperatura, eliminando así la posibilidad de que los animales queden sin agua durante periodos prolongados de tiempo debido a un despiste humano o a una avería.

- Control de la valla electrificada. Esto posibilita la monitorización y control del pastor eléctrico en cualquier momento y desde cualquier sitio.

- Histórico de cada zona de pastoreo. El responsable de la explotación puede planificar la próxima zona de pastoreo, en base a un historial almacenado en el sistema, permitiéndole así maximizar la productividad del terreno.

- Detección de anomalías y envío de alertas. Se notifica al ganadero en caso de detectar que algún tanque de agua baje de un nivel preestablecido, algún dispositivo de localización tenga la batería baja o que se produzca una alarma.

- Detección de animales que se alejan del grupo. Se acortará el tiempo de búsqueda reduciendo la probabilidad de pérdida de animales.

- Sistema desasistido. Se considera hardware de bajo consumo que permite el uso de dispositivos pequeños, con baterías que proporcionan gran autonomía y transmisiones que cubren un área muy extensa.

Se muestra una representación general del sistema en la figura 1.3. Los nodos finales se comunican con el servidor central de IoT por medio de las puertas de enlace. Este servidor es la parte encargada de comunicarse, de forma bidireccional, con el servidor de la aplicación.

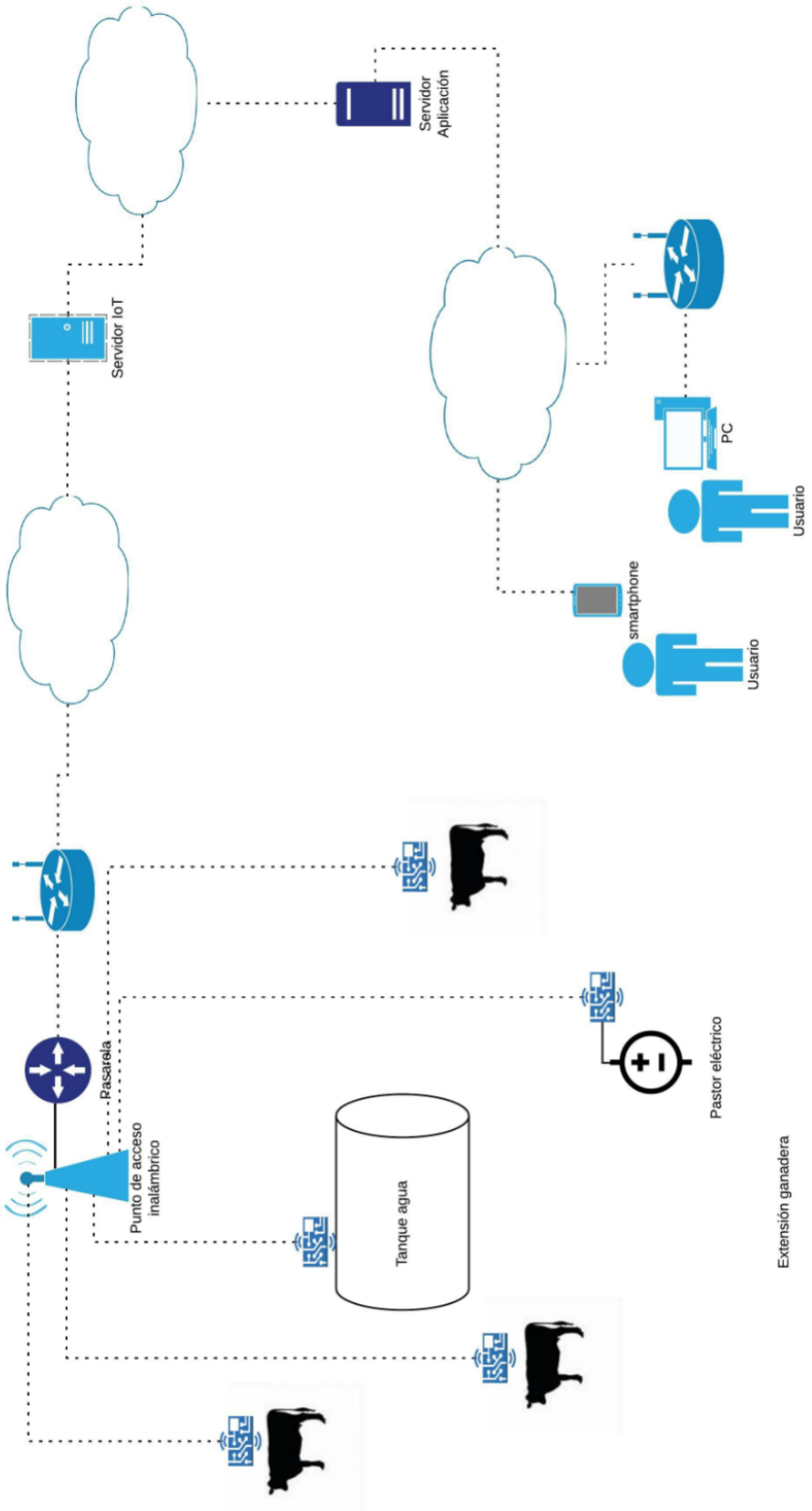


Figura 1.3: Diagrama estructural del sistema.

1.7 Estructura de la memoria

La memoria de este Trabajo Fin de Grado (TFG) está dividida en los capítulos detallados a continuación:

1. Introducción. En este capítulo se ha explicado el origen del proyecto y los objetivos del mismo y se mencionan, de forma general, las tecnologías existentes para abordar los problemas planteados. También se detalla la metodología de desarrollo del TFG e incluye una breve descripción de las funcionalidades implementadas.

2. Estado del arte de las tecnologías. En este capítulo se plasma un resumen de la investigación sobre la situación de las tecnologías que actualmente sirven para solucionar el problema abordado en el proyecto y algunos hitos relevantes. También aquí se refleja la base tecnológica sobre la que se ha realizado en este TFG.

3. Nodos finales. Capítulo en el que se detallan el montaje, el desarrollo y la configuración de cada dispositivo utilizado, incluyendo una descripción de sus componentes y discusión sobre la elección.

4. Red de comunicaciones de internet de las cosas. Se trata en este capítulo sobre la red utilizada por los dispositivos finales para su comunicación con el servidor de IoT y este, a su vez, con el servidor de la aplicación.

5. Servidor. Se analizan en este capítulo el hardware y el software sobre los que corre la aplicación servidora, detallando sus características principales y explicando como realizar los pasos de mayor complejidad.

6. Conclusiones y líneas futuras. Se expone en este capítulo, un análisis del desarrollo del proyecto, así como una exposición de resultados. También se detallan aquí los costos del sistema y las posibles mejoras del mismo.

Capítulo 2

Estado del arte de las tecnologías

En este capítulo se realiza la presentación y el análisis del estado del arte en el ámbito relativo al proyecto, así como las tecnologías y técnicas necesarias para el diseño del sistema planteado. Básicamente, se expone la situación actual en sistemas de monitorización y control de las explotaciones de ganadería extensiva.

2.1 Comunicaciones

Las comunicaciones con los nodos finales son a largas distancias y los dispositivos deben tener la suficiente autonomía para obtener datos de los sensores, codificarlos y enviarlos con el menor consumo posible, maximizando así la duración de sus pequeñas baterías. Para conseguir transmisiones de datos de forma inalámbrica se deben considerar las diferentes tecnologías existentes en el mercado, compararlas y decidir cuál utilizar.

2.1.1 Redes LPWAN

Las redes Lower Power Wide Area Network (LPWAN) [9] son un tipo de redes de telecomunicaciones inalámbricas, diseñadas para para ser utilizadas en comunicaciones de largo alcance con una baja velocidad de transmisión y los atributos tecnológicos que detallan a continuación:

- Largo alcance. El alcance operativo de esta tecnología varía desde unos cientos de metros en áreas urbanas a más de 15 km en entornos rurales.
- Baja potencia. Transceptores optimizados para un bajo consumo de batería usando pequeñas baterías.

	WiFi	**BLE**	**Zigbee**	**LoRa**	**Sigfox**	**4G/5G**
Transferencia de datos	10 Gbps	1.4 Mbps	250 Kbps	50 Kbps	1 Kbps	10 Gbps
Consumo de energía	Alto	Bajo	Muy Bajo	Muy bajo	Muy bajo	Medio
Costo	Bajo	Medio	Bajo	Bajo	Bajo	Medio
Frecuencias	2.4 GHz y 5 GHz	2.4 GHz	2.4 GHz (EU)	868 MHz (EU)	868 MHz (EU)	3 - 30 GHz
Alcance	100 m	100 m	20 m	20 km	30 km	1 km

Tabla 2.1: Tabla comparativa de las tecnologías más utilizadas para dispositivos IoT

- Bajo costo. Gracias a utilizar protocolos ligeros y hardware sencillo, se reducen los costos de los dispositivos.

2.1.2 Tecnologías más utilizadas en internet de las cosas

En la tabla 2.1 se muestran las tecnologías más utilizadas en IoT, algunas de ellas pertenecientes a las LPWAN, y las características más importantes en el momento de su elección. Después de su análisis, es de destacar, que LoRa [10] y Sigfox [11] tienen mayor alcance en sus comunicaciones y menores consumos, aunque también son las tienen la tasa de transferencia más baja.

Después del análisis de las tecnologías más utilizadas, se concluye que LoRa es la tecnología más adecuada para el desarrollo del proyecto y, por tanto, se decide utilizarla como tecnología de telecomunicaciones de los nodos finales con el servidor de IoT.

LoRa es una tecnología inalámbrica localizada dentro de las LPWAN que, a pesar de tener una baja velocidad de transferencia, es la que mejor se adapta a las necesidades del proyecto; debido a la posibilidad que ofrece de realizar transmisiones de datos a largas distancias, su bajo consumo y la posibilidad de adquirir dispositivos a bajos precios. A esto hay que sumarle, que a diferencia de Sigfox, 4G [12] y 5G [13] no depende de un operador de red global, si no que permite la instalación puertas de enlace libremente.

2.1.3 LoRa

Este tipo de tecnología utiliza un tipo de modulación en radiofrecuencia patentado por Semtech [14], actualmente administrado por “LoRa Alliance” [10], denominado Chirp Spread

Spectrum, muy utilizado en comunicaciones espaciales y militares desde hace décadas.

LoRa puede lograr comunicaciones fiables a largas distancias (hasta 20 km con condiciones favorables), con alta tolerancia a las interferencias, utilizando las bandas Industrial Scientific & Medical (ISM), reservadas internacionalmente para el uso de energía de radiofrecuencia para fines industriales, científicos y médicos distintos de las telecomunicaciones. A pesar de la intención de las asignaciones originales, en los últimos años los usos de más rápido crecimiento de estas bandas han sido para sistemas de comunicaciones inalámbricas de corto y largo alcance, ya que estas bandas, a menudo, están aprobadas para tales dispositivos, que se pueden usar sin licencia.

Los principales parámetros de comunicación LoRa son:

- Canal. Frecuencia central que representa la transmisión.
- Spreading Factor (SF). Define el número de bits usados para codificar un símbolo (clasificado entre 7 - 12). A mayor SF, menor velocidad de transferencia pero mayor inmunidad a interferencias.
- Coding Rate (CR). Indica la forma de codificar para corrección de errores.
- Bandwidth (BW). Indica el ancho de banda del canal que va a utilizar.

Debido a la ortogonalidad de modulación y el uso de espectro ensanchado, se pueden decodificar múltiples señales a pesar de usar la misma frecuencia. Esta característica dota a LoRa de “canales virtuales” [15].

2.1.4 LoRaWAN

Long Range Wide Area Network (LoRaWAN) es un protocolo de red que utiliza LoRa, diseñado para permitir que los dispositivos de baja potencia se comuniquen con aplicaciones conectadas a Internet a través de conexiones inalámbricas de largo alcance. LoRa se puede asignar a la segunda y tercera capa del modelo OSI y se implementa sobre la modulación LoRa o FSK en bandas de radio ISM. Los protocolos LoRaWAN están definidos por LoRa Alliance y formalizados en la especificación LoRaWAN. Se representa gráficamente su arquitectura en la figura 2.1.

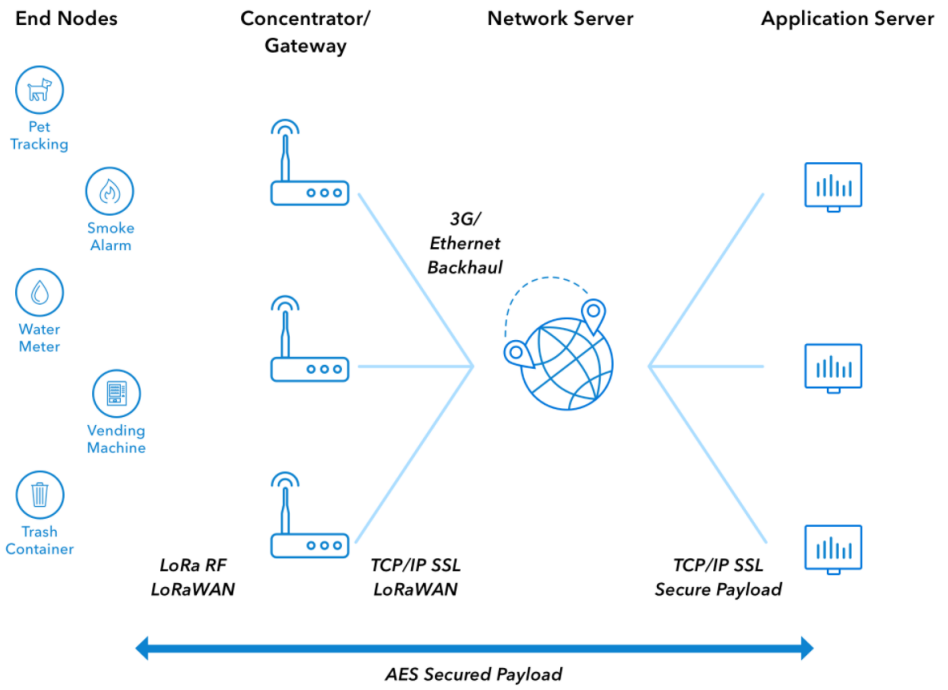


Figura 2.1: Esquema de elementos de una red LoRaWAN. Los nodos finales intercambian información con las pasarelas, que se comunican con el servidor de red central y éste con las aplicaciones.

En una transmisión LoRaWAN se tienen en cuenta los siguientes componentes principales:

- Dispositivo final o nodo. Es un elemento hardware con un dispositivo de comunicación de bajo consumo integrado.
- Puerta de enlace. Son los elementos que se comunican con los de dispositivos finales.
- Servidor de red. Son los servidores que enrutan mensajes desde los dispositivos finales a la aplicación correcta y viceversa.
- Aplicación. Es una pieza de software que se ejecuta en un servidor.

Tipos de enlace

- Enlaces ascendentes.
  Se conocen como enlaces ascendentes a las transmisiones de datos desde los nodos finales. Para conseguir comunicaciones eficientes se deben cumplir las siguientes recomendaciones:

  - Optimizar la carga útil de la transmisión.
  - Controlar el intervalo de envío de mensajes, optimizando el envío de datos para hacer las transmisiones solo cuando sea necesario.
  - La velocidad de transmisión de datos debe ser lo más rápida posible para minimizar el tiempo de transmisión. En caso de nodos estáticos se recomienda utilizar Adaptive Data Rate (ADR), de esta forma la red optimizará automáticamente su velocidad de datos.

- Enlaces descendentes.
  Se conocen como enlaces descendentes, a las transmisiones desde el servidor central a los nodos finales. Estos mensajes, pueden ser utilizados para la configuración y control de los dispositivos de forma remota. La periodicidad con la que estos mensajes pueden ser enviados va ser determinada por la clase implementada por cada dispositivo.

Para más información sobre la implementación de LoRaWAN, ver el apéndice A.2.

Principales hitos tecnológicos

A finales del 2019 se lanzó el FossaSat-1 [16], un picosatélite de código abierto lanzado a través del Electron Rocket [17] con un repetidor LoRa integrado diseñado para comunicaciones de IoT, con unas dimensiones de 5 x 5 x 5 cm y un peso de 250 g. Posteriormente, se confirma que el FossaSat-1 fue lanzado con éxito pero las antenas no se desplegaron correctamente, por lo que la señal emitida era débil y no podía ser recibida por las pequeñas estaciones terrestres. En la figura 2.2 aparece la imagen del picosatélite.

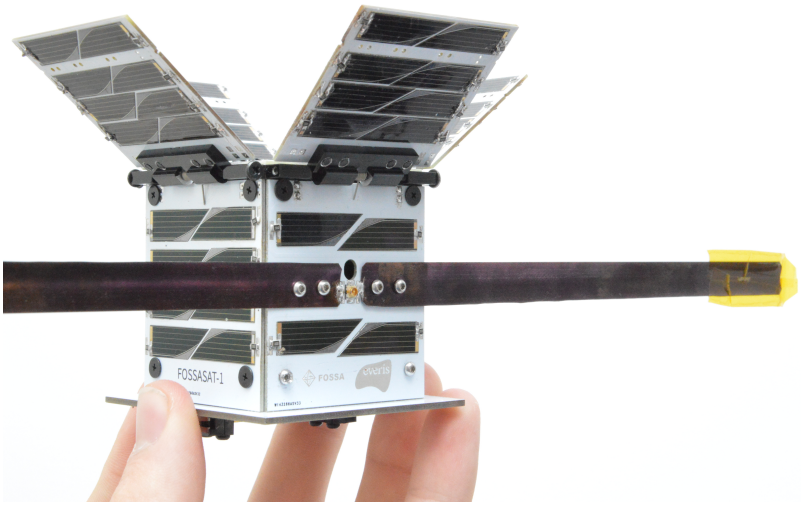


Figura 2.2: En la figura se aprecian las reducidas dimensiones del picosatélite FossaSat-1.

Esta tecnología también es utilizada en la construcción de globos meteorológicos, que son globos aerostáticos [18] construidos con látex e inflados con helio o hidrógeno. Estos globos alcanzan unos 30 km de altura, distancia a la que que suelen explotar y empezar su descenso en paracaídas. Son comúnmente utilizados para realizar observaciones de las condiciones atmosféricas en tiempo real, así como para efectuar modelos numéricos de predicción del tiempo, soliendo llevar incorporados sensores para medir la humedad, temperatura, velocidad y dirección del viento, así como dispositivos de localización Global Navigation Satellite System (GNSS).

En un proyecto de aficionados, en Zaragoza [19], se lanzaron varios de estos globos transportando diversos experimentos científicos. Como se refleja en el mapa de la figura 2.3, uno de los globos, equipado con LoRaWAN logró comunicarse con la antena Lorix One [20] del Instituto Superior de Ingeniería de Lisboa, ubicada en la capital portuguesa, a 741 km de distancia utilizando una potencia de transmisión de 14 dBm.

Figura 2.3: En el mapa aparecen representados por globos los dispositivos utilizados en el concurso de Zaragoza, 54 horas después del inicio del mismo.

2.2 Hardware

Para la creación de los nodos finales, es necesario utilizar sensores y actuadores conectados a un microcontrolador central y a un módulo de transmisión LoRa. Estos dispositivos serán alimentados con pequeñas baterías, a excepción del que controla la valla eléctrica, que va alimentado con el mismo generador que la propia valla.

Tanto Arduino [21] como Heltec [22] proporcionan placas de desarrollo de IoT, diseñadas y producidas por ellos mismos, altamente integradas. Los productos de Heltec están basados en ESP32 [23] y los de Arduino en SAMD21 Cortex [24]. Además, ambos incorporan el módulo SX127x [25] para las comunicaciones por LoRa y, dependiendo de las necesidades del proyecto, se pueden conseguir chips con funciones Wi-Fi, BLE, sistema de gestión de batería, diferente número de pines de entrada y salida, módulos GNSS e incluso con pequeñas pantallas OLED incorporadas. Son buena opciones cuando el objetivo es diseñar y desarrollar dispositivos personalizados.

Otra opción muy interesante, aunque menos didáctica y flexible, es la utilización de productos finales terminados, donde los dispositivos incorporan los microcontroladores y los distintos sensores que se necesitan para satisfacer las necesidades de cada caso de uso. Para estas opciones, existen en el mercado empresas como Dragino [26] o Tektelic [27] que proporcionan estos productos ensamblados y programados, que envían la información mediante LoRaWAN. Estos dispositivos suelen ser configurables mediante este protocolo, utilizando mensajes descendentes; o conectados a la computadora, utilizando los códigos proporcionados por el fabricante en la ficha técnica de cada dispositivo.

2.2.1 Monitorización de nivel y temperatura del tanque de agua

Para la monitorización del nivel de los tanques de líquidos [28], existen multitud de métodos, divididos en dos tipos generales:

- Sensores de nivel de punto. Utilizados para obtener la altura de un líquido en un determinado nivel preestablecido. Existen sensores con funcionamientos muy distintos: de flotador, ultrasónicos, de capacitancia, etc.

- Sensores de nivel continuos. Son más sofisticados y pueden realizar el seguimiento del nivel de todo un sistema. Indicado para sistemas complejos, donde no sea suficiente con medir el agua en solo un punto

2.2.2 Control del pastor eléctrico

Un pastor eléctrico [29] es un dispositivo que genera impulsos de alto voltaje entre sus dos terminales de salida. En esta ocasión: una valla eléctrica y la toma de tierra. El impulso es breve, así que, el animal no sufre ningún daño, pero si experimenta una sensación molesta, que impide que este se arrime a la cerca.

Para poder controlarlo de forma remota, se utiliza un microcontrolador sencillo con conectividad LoRa, que controla un relé que hará las veces de interruptor.

2.2.3 Localización del ganado

Se han valorado varios métodos para saber la posición del ganado.

- Colocación de arcos Radio Frequency Identification (RFID) [30]. De esta forma se registra en el sistema los arcos que ha cruzado cada animal con su hora y fecha exacta. Para la implantación de este sistema, se requiere una gran inversión, debido a la instalación de numerosos arcos RFID, a los que hay que alimentar. Otra de las desventajas de este sistema es que existe una elevada dificultad para garantizar que los animales solo cambian de zona pasando por estos arcos. La gran ventaja de esta solución es la posibilidad de utilizar etiquetas RFID pasivas, prescindiendo de la utilización de baterías.

- Utilización de sistemas Local Positioning System) (LPS). Estos sistemas, de gran utilidad en localización de interiores, se construyen instalando balizas activas sobre el terreno. Inviable debido a la gran extensión y a la compleja geometría del mismo.

- Sistemas de localización GNSS. Utilizando sistemas de posicionamiento global se evita cualquier tipo de instalación en la finca. El principal inconveniente de este sistema, es la dependencia de las pequeñas baterías que alimentan a cada dispositivo.

Para llevar a cabo este proyecto, se ha decidido utilizar dispositivos de localización por GNSS, debido a la gran extensión de terreno en el que es necesario localizar a los animales y la posibilidad de que los animales se salgan de los límites establecidos.

2.3 Servidor

En cuanto a los servidores de aplicaciones, se podría optar por un servidor en la nube, en el que el desarrollador se despreocuparía del mantenimiento, disponibilidad y escalabilidad de los recursos o por crear un servidor físico en la red local.

En el desarrollo de este proyecto, debido a que se trata de un proyecto académico, se ha decidido montar un servidor físico sobre una Single Board Computer (SBC). la Raspberry Pi 4 [31], aunque su extensión a cualquier otro tipo de servidor es trivial.

2.3.1 Sistema Operativo

Para la instalación del sistema operativo en la Raspberry Pi® existen multitud de posibilidades, siendo Raspberry Pi OS [32] el sistema por excelencia y, casi por defecto, para este tipo de placas.

Raspberry Pi OS es una distribución del sistema operativo GNU/Linux basado en Debian, y por lo tanto libre para la SBC Raspberry Pi®. Debido al número y tipo de herramientas que integra por defecto, es una distribución óptima para la realización de tareas generales.

A parte del SO anterior, existen muchos otros para instalar en la SBC, dependiendo del modelo del dispositivo y sus prestaciones, así como, de la aplicación a la que vaya destinado.

Para el desarrollo de este sistema se ha decidido instalar Home Assistant Operating System (HassOS) [33], un SO basado en Buildroot LTS [34], optimizado para alojar Home Assistant (HA) que es un software de automatización de dispositivos gratuito y de código abierto. Tanto HA como sus extensiones de software están escritas en Python, siendo su enfoque principal el control local y la privacidad [35].

2.3.2 Herramienta de desarrollo

Aunque los lenguajes dominantes en el mundo del IoT siguen siendo C, C++, Java, JavaScript, Phyton, etc., en 2013 comenzó su andadura Node-RED [36]. Esta herramienta, en principio, era usada para visualizar y manipular flujos entre temas Message Queuing Telemetry Transport (MQTT) pero, después de una gran evolución, salió a la luz siendo uno de los proyectos de la JS Foundation [37].

Node-RED es una herramienta de programación para conectar dispositivos hardware, APIs y servicios en línea que proporciona un editor de flujo basado en navegador. Esta herramienta facilita la conexión de flujos mediante la amplia gama de nodos de la paleta o mediante la creación de nodos propios utilizando JavaScript.

2.3.3 Base de datos

En la actualidad, hay multitud de gestores de bases de datos para poder utilizar en cada aplicación. pero cuando las necesidades a cubrir son el almacenamiento de series temporales y consultas en tiempo real, las Time Series Database (TSDB) son las que ofrecen más ventajas

con respecto a las bases de datos relacionales. Están optimizadas para una gran velocidad de lectura y escritura en tiempo real, así como para una realización de operaciones de filtrado basadas en marcas temporales. Se componen generalmente de una marca temporal y una tupla de pares clave-valor que representan cada una de las variables dependientes.

En la gráfica de la figura 2.4 se muestra la evolución de los tipos de DBMS [38] más utilizados en el mundo. En la gráfica de los últimos 24 meses se aprecia el despunte de la utilización de estos gestores frente al estancamiento otros, como los gestores de BBDD relacionales.

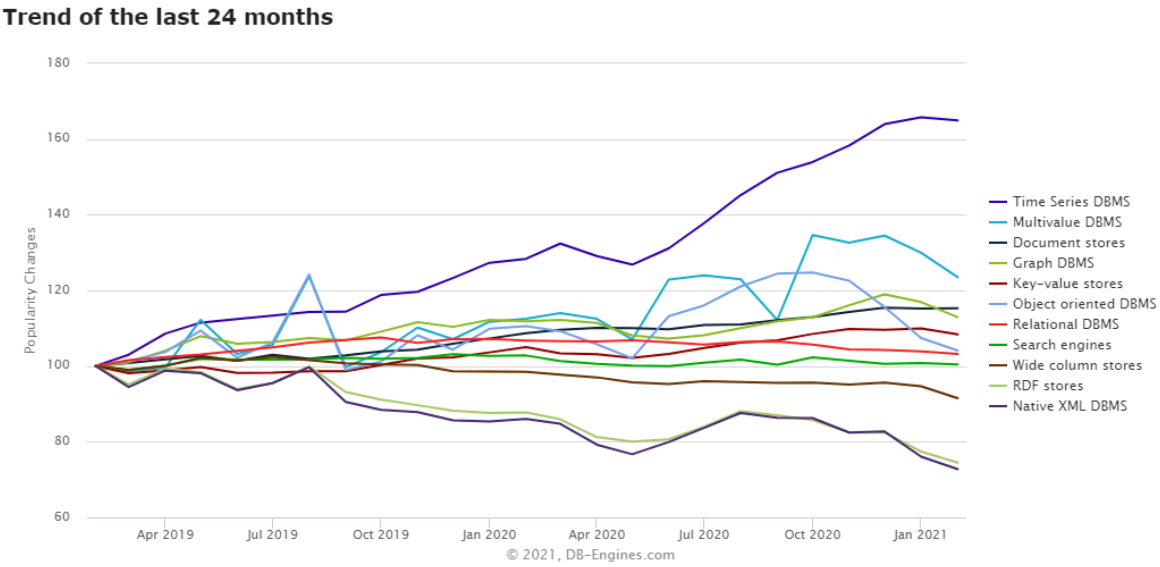


Figura 2.4: Evolución de la popularidad de los principales tipos de DBMS.

InfluxDB [39], un gestor TSDB de código abierto desarrollada por InfluxData [40]. Este gestor, escrito en Go, está optimizado para el almacenamiento y la recuperación rápida de alta disponibilidad en campos como la monitorización de operaciones, métricas de aplicaciones, datos de sensores de IoT y análisis en tiempo real.

Otros proyectos de código abierto usados en IoT son:

- OpenTSDB [41]. Es una base de datos distribuida y escalable de series de tiempo.
- Prometheus [42]. Almacena todos los datos de muestra recopilados en la base de datos de memoria por series de tiempo y guarda regularmente los datos en el disco duro.
- TimescaleDB [43]. BBDD SQL de series de tiempo basada en PostgreSQL [44].

HA usa SQLAlchemy [45], que es un Object Relational Mapper (ORM). Esto significa que puede usar de forma nativa cualquier backend Structured Query Language (SQL) que sea soportado por SQLAlchemy.

En este sistema, por simplicidad, se utiliza SQLite [46] como motor de base de datos predeterminado, debido que no requiere ninguna configuración adicional para su uso y es suficiente para la pequeña cantidad de datos que es almacenada.

2.3.4 Software de visualización

A menudo es necesario realizar representaciones gráficas de la información y los datos recolectados mediante el uso de elementos visuales, como gráficos y mapas. La visualización de datos ofrece una manera accesible para detectar y comprender las tendencias, los valores atípicos y los patrones en los datos.

Para ello, existen potentes herramientas como Grafana [47]. Esta herramienta está especialmente diseñada para visualizar datos de serie temporales. A partir de una serie de datos recolectados, se obtiene un panorama gráfico de la situación de un conjunto de sensores, empresa u otro tipo de dato.

Por simplicidad, en el nivel de madurez que se encuentra el proyecto, se utilizan las propias tarjetas de visualización que provee HA.

2.4 Soluciones existentes actualmente para la ganadería extensiva

Las dos empresas del mercado español con soluciones interesantes diseñadas para la ganadería extensiva son Digitanimal S.L. y Accent Advanced Systems S.L., tal y como se ha mencionado en la sección 1.3.

Digitanimal S.L. utiliza Sigfox, Narrowband for the Internet of Things (NB-IoT) o Global System for Mobile communications (GSM) para las comunicaciones con el servidor de IoT. Las soluciones que ofrece son para controlar el ganado, basadas en localización GNSS. Permite acceder a los datos en tiempo real a través del navegador web o a través de la aplicación disponible para teléfono inteligente o para tableta. Además, dispone de API posibilitando la integración con otras aplicaciones, ofreciendo datos en bruto en diferentes formatos.

Accent Advanced Systems S.L. se dedica a multitud de soluciones IoT. Recientemente se ha incorporado al mundo de la ganadería extensiva, ofreciendo al ganadero un producto para traquear los movimientos de los animales y analizar los principales parámetros de los mismos. Las comunicaciones con el servidor IoT las realiza mediante NB-IoT.

Además de las empresas anteriores, en este campo, existen varios proyectos en estudio. Uno de los más innovadores, que todavía no salió al mercado, es el perteneciente al Grupo Operativo Siega [48], que combinará el seguimiento de los pastos por teledetección con el posicionamiento del ganado. Otro proyecto, es el conocido como GELOB [49], que permitirá geolocalizar y monitorizar al ganado que pasta por el monte para, entre otros objetivos, detectar y prevenir ataques de lobos.

2.4.1 Crítica al estado del arte

Se detecta en el estudio de los productos existentes que, aún siendo de alta calidad y cubriendo gran parte de las necesidades del ganadero, se centran en el control del ganado, sin abordar la monitorización y la gestión de las instalaciones, asuntos de vital importancia para las empresas a las que van destinados.

En el mercado, existe una amplia gama de soluciones para entornos industriales comunes, pero muy reducida cuando se necesita realizar comunicaciones a largas distancias, en zonas sin cobertura y los dispositivos deben ser alimentados por pequeñas baterías.

Esta carencia, supone un trastorno a los ganaderos, debido a que les empuja a contratar los servicios de otra empresa para gestionar estos aspectos, implicando el aprendizaje y mantenimiento de varios sistemas.

Otro de los inconvenientes que se detecta en algunos productos es que no utilizan tecnologías libres, sino que utilizan un operador de red global. Para la utilización de este operador, suele ser necesario contratar una suscripción periódica por cada dispositivo, gasto que repercute directamente al ganadero.

2.4.2 Propuesta

Se propone diseñar un sistema para la monitorización y control de la ganadería completa, evitando tener múltiples aplicaciones descentralizadas. Este sistema no solo ofrece al ganadero la monitorización de la posición del ganado, si no que incluye un sistema para monitorizar el nivel de llenado y temperatura de los tanques de agua con envío de alertas de nivel bajo.

Además, se incluye un sistema de control remoto de la vaya electrificada.

Las comunicaciones con el servidor IoT se realizarán mediante LoRaWAN y el control del ganado mediante dispositivos de localización GNSS.

También se propone mejorar el coste económico de las soluciones existentes, debido a que, en la solución propuesta, no es necesario utilizar los servicios de un operador de red global, como los utilizados por las dos empresas mencionadas en la sección 2.4, si no que se utiliza LoRa [50], que es una tecnología libre, eliminando de esta forma cualquier tipo de pago periódico.

Capítulo 3

Nodos finales

PARA monitorizar y controlar los diferentes parámetros de la explotación, es necesario contar con elementos hardware en las propias instalaciones de la misma, algunos de ellos, como los localizadores, colocados en los propios animales. Estos dispositivos, conocidos como nodos finales, están formados por sensores y actuadores conectados a microcontroladores que se comunican de forma inalámbrica, haciendo uso de la especificación LoRaWAN, con el servidor central de IoT.

En este capítulo, se hará referencia al montaje, desarrollo y configuración de cada dispositivo final dividido y organizado por casos de uso.

3.1 Monitorización del tanque

Conocer la cantidad de agua remanente en el tanque es un asunto primordial para el ganadero. El vaciado completo del mismo durante periodos prolongados de tiempo podría suponer la fuga o la muerte de las reses. En la figura 3.1 se muestra el tanque de agua perteneciente a la explotación para la que es diseñado el sistema.

En este caso, debido a que el agua no produce turbulencias ni vapor en la superficie, lo que dificultaría las mediciones, y a que el tanque es móvil, para que pueda ser llevado a zonas de carga de agua, es necesario utilizar un sistema sencillo, robusto y resistente al agua. Por ello, se decide utilizar un sensor de ultrasonidos estanco JSN-SR04T [51] y un sensor de temperatura DS18B20 [52], ambos conectados a un microprocesador Arduino MKR 1310 [53].

Figura 3.1: Tanque de 3 500 litros situado en la explotación.

Para calcular el contenido de cada tanque se mide la distancia desde la parte superior del mismo a la superficie del agua. Esa medida, junto con la obtenida por el sensor de temperatura, se codifica y se envía mediante LoRaWAN al servidor IoT. Datos que, posteriormente, serán decodificados y procesados.

3.1.1 Microcontrolador

Para procesar los datos recolectados por los sensores y enviarlos al servidor central, es necesario conectarlos a un microcontrolador y a un transmisor LoRa. Para ello, se ha utilizado el módulo Arduino MKR WAN 1310, que es un microcontrolador de código abierto con conectividad LoRa y bajo consumo de energía. En la tabla 3.1 se muestran sus características principales.

3.1.2 Sensor JSN-SR04T

Para medir la distancia desde la parte superior del tanque a la superficie del agua se coloca el sensor, que aparece en la fotografía de la figura 3.2, en la parte superior del tanque, centrado en los dos ejes paralelos a la superficie. El JSN-SR04T [51] es un sensor ultrasónico estanco. Se muestran otras características del sensor en la tabla 3.2.

Arduino MKR WAN 1310	
Microcontrolador	SAMD21 Cortex ®-M0
Módulo de radio	CMWX1ZZABZ
Voltaje circuito	5 V
Consumo (sleep mode)	104uA
Pines digitales E/S	8
Pines PWM	13
Memoria Flash	256 KB
Frecuencia portadora	433/868/915 MHz
Dimensiones	67.64 x 25 x 12 mm

Tabla 3.1: Características del Arduino MKR WAN 1310.

Módulo JSN-SR04T	
Alimentación	3.3 a 5V
Consumo (sleep mode)	5 mA
Consumo en funcionamiento	30 mA
Rango distancia	20 a 600 cm
Resolución	1 mm
Precisión	± 1 cm
Temperatura de funcionamiento	$-20 \sim +70$℃
Ángulo de detección	< 70º
Frecuencia de emisión acústica	40 KHz
Dimensiones	42 x 29 x 12 mm

Tabla 3.2: Características del módulo JSN-SR04T.

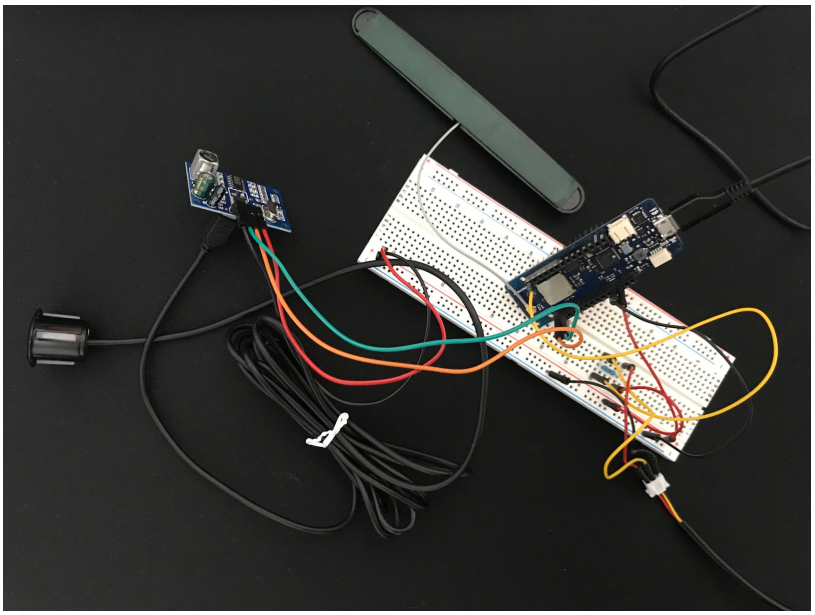


Figura 3.2: El sensor JSN-SR04T conectado al microcontrolador Arduino MKR 1310.

Sensor DS18B20	
Alimentación	3 a 5.5V
Consumo (sleep mode)	750 nA
Consumo en funcionamiento	1.5 mA
Resolución	0.2 ℃
Precisión (Rango -10℃ $\sim +85$℃)	± 0.5℃
Temperatura de funcionamiento	$-55 \sim +155$℃
Dimensiones	50 x 5 x 5 mm

Tabla 3.3: Características del sensor DS18B20.

3.1.3 Sensor DS18B20

Este sensor de temperatura proporciona la salida mediante un bus de comunicación digital que puede ser leído con las entradas digitales de Arduino, disponiendo de un rango amplio de medición y una precisión válida para el caso de uso planteado. Emplea un bus de comunicación denominado 1-Wire cuya principal ventaja es que necesita un único conductor para realizar la comunicación (sin contar el conductor de tierra). Los dispositivos pueden ser alimentados directamente por la línea de datos o mediante una línea adicional requiriendo una resistencia de pull-up en la línea de datos.

Se muestra una fotografía del montaje en la figura 3.3 y el resultado de un experimento en la figura 3.4.

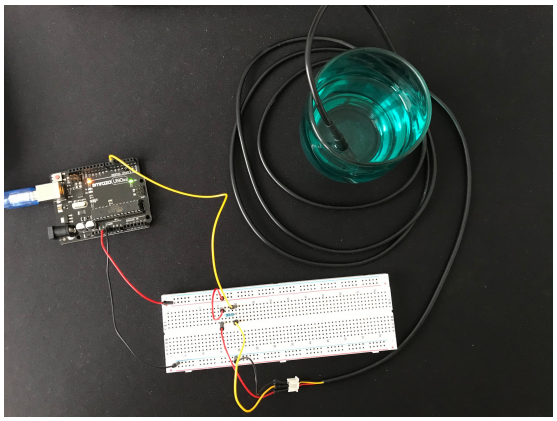


Figura 3.3: Prototipo del medidor de temperatura para el tanque de agua donde el sensor DS18B20 está conectado a un microcontrolador.

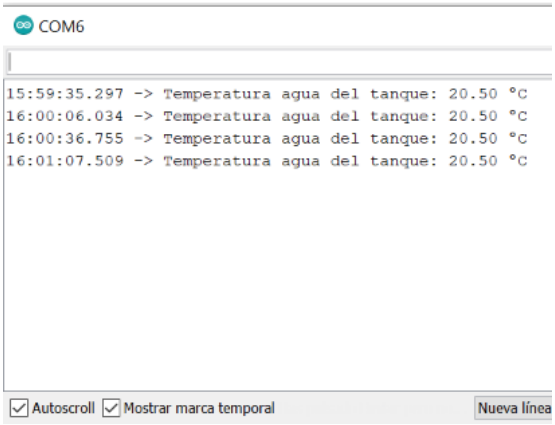


Figura 3.4: Experimento que muestra la estabilidad de las mediciones de temperatura realizadas.

3.2 Geolocalización de ganado

Esta funcionalidad permite al ganadero saber dónde se encuentran los animales en cada momento y visualizar sus recorridos diarios. En la figura 3.5 se muestra una fotografía de una pequeña parte de la ganadería en medio de la grande extensión de terreno donde pastan.

Figura 3.5: Ganado pastando libremente en una gran extensión de terreno perteneciente a la explotación.

Para la localización del ganado se han utilizado distintos modelos de dispositivos de localización Global Navigation Satellite System (GNSS), posibilitando así la comparación de éstos, el análisis y la obtención de conclusiones.

3.2.1 HTCC-AB02S

Uno de los nodos finales está basado en el microcontrolador de bajo consumo HTCC-AB02S [54], que aparece en la figura 3.6 conectado a una batería externa para la realización de pruebas de campo. Este dispositivo fabricado por Heltec [55] incorpora una pequeña pantalla OLED, dispone de un módulo GNSS Air530 [56] integrado y conectividad LoRaWAN.

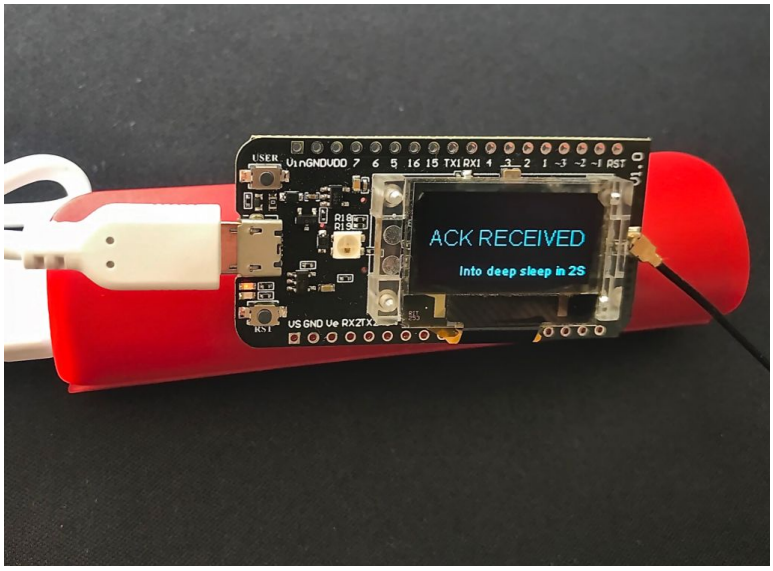


Figura 3.6: Placa de desarrollo HTCC-AB02S conectada a una batería externa.

3.2.2 Dragino LGT92-LI

El localizador basado en GNSS Dragino LoRaWAN LGT-92-LI [57], que aparece en la figura 3.7, está formado por un MCU STM32L072 [58] y un módulo LoRa SX1276 [25]. Este modelo se proporciona alojado en una carcasa protectora e incluye una batería de litio recargable de 1000 mAh.

Figura 3.7: Fotografía del dispositivo Dragino LGT-92 LI donde se aprecia el pequeño tamaño del dispositivo al lado de un bolígrafo.

3.2.3 Dragino LGT92-AA

Comparte la misma base que el Dragino LG92-LI3.2.2 sin incluir carcasa ni batería. Este modelo, que se muestra en la figura 3.8, está diseñado para ser alimentado con 2 pilas AA.

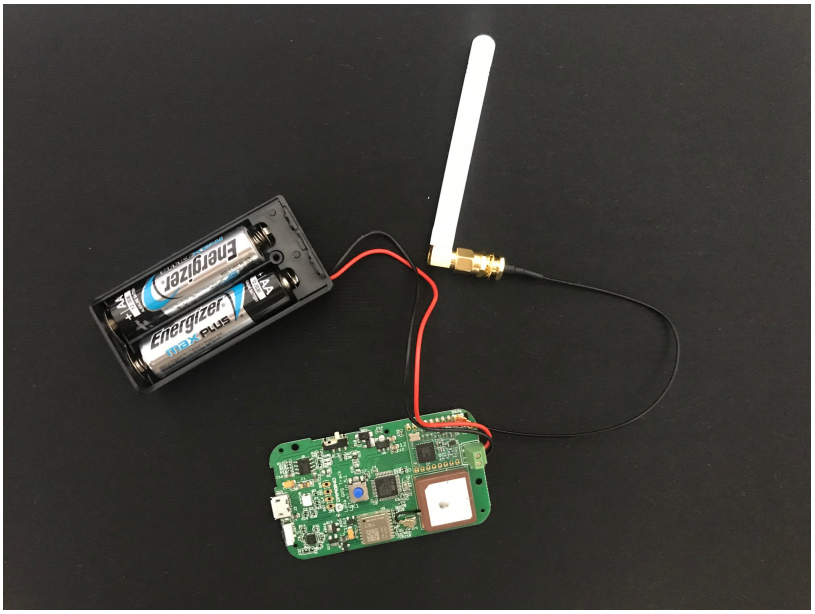


Figura 3.8: Dispositivo Dragino LGT-92 AA.

3.2.4 Configuración de dispositivos de forma remota

Es interesante poder controla los dispositivos a distancia, evitando el desplazamiento a sitios de difícil acceso donde estén instalados y conectarlos a un ordenador para reajustar su configuración.

Para solucionar este problema, se hará mediante el envío de mensajes descendentes vía LoRaWAN, haciendo uso de comandos AT, que están definidos, en caso de ser nodos finales programados de fábrica, en la ficha técnica del dispositivo.

3.2.5 Comparación y análisis de los distintos dispositivos de localización utilizados

Después de las pruebas y análisis de los distintos dispositivos de geolocalización utilizados, se considera que los óptimos para llevar a cabo el proyecto son los Dragino LGT92-AA. Siendo los de menor consumo en modo dormido y utilizando el módulo L76-L [59]. Este módulo es de mayor precisión que el Air530 debido a la habilitación optimizada de varios sistemas Global Navigation Satellite System (GNSS) que, generalmente, aumenta la cantidad de satélites visi-

	CubeCell GPS-6502	Dragino LGT92-LI	Dragino LGT92-AA
Consumo (sleep mode)	50uA	77uA	17uA
Consumo GPS	31 mA	38 mA	38 mA
Precisión	< 10 m	< 2.5 m	< 2.5m
Consumo tx	105 mA	150 mA	150 mA
Potencia máx. tx	22 dB	24 dB	24 dB
Arranque en frío	60 s dB	< 35 s dB	< 35 s dB
Arranque en caliente	35 s	< 30 s dB	< 35 s dB
Rango temperatura	-40 ~ +80ºC	-40 ~ +85ºC	-40 ~ +85ºC
GNSS	Air530	L76-L	L76-L
Batería incluida	No	Si	No
Botón Alarma	No	Si	Si
Acelerómetro	No	Si	Si
Pines I/O	Si	No	No
Dimensiones	55.9 x 27.9 x 9.5 mm	85 x 48 x 15 mm	75 x 40 x 10 mm

Tabla 3.4: Tabla comparativa de los distintos dispositivos GPS utilizados en el desarrollo del proyecto.

bles, reduce el tiempo hasta la primera corrección y aumenta la precisión de posicionamiento, especialmente cuando se trata de entornos difíciles. Los modelos Dragino ofrecen, además, una mayor potencia de transmisión y un menor tiempo de arranque, con la ventaja de que llevan incorporados de serie el soporte de la alimentación.

El módulo HTCC-AB02S, es de gran utilidad para la creación de prototipos y pruebas de campo, posibilitando la visualización en tiempo real del estado de la conexión y las coordenadas de posicionamiento global en su pequeña pantalla OLED. Este dispositivo tiene la ventaja de disponer de pines de entrada y salida que son necesarios en otros casos de uso, en los que es necesario conectar algún sensor y/o actuador al módulo.

3.3 Control de valla electrificada

Para la monitorización y control de la valla electrificada se utiliza un microcontrolador HTCC-AB01 [60] con conectividad LoRa, conectado a un pastor eléctrico a través de un relé intermedio, que funciona como un interruptor controlado por el microcontrolador. .

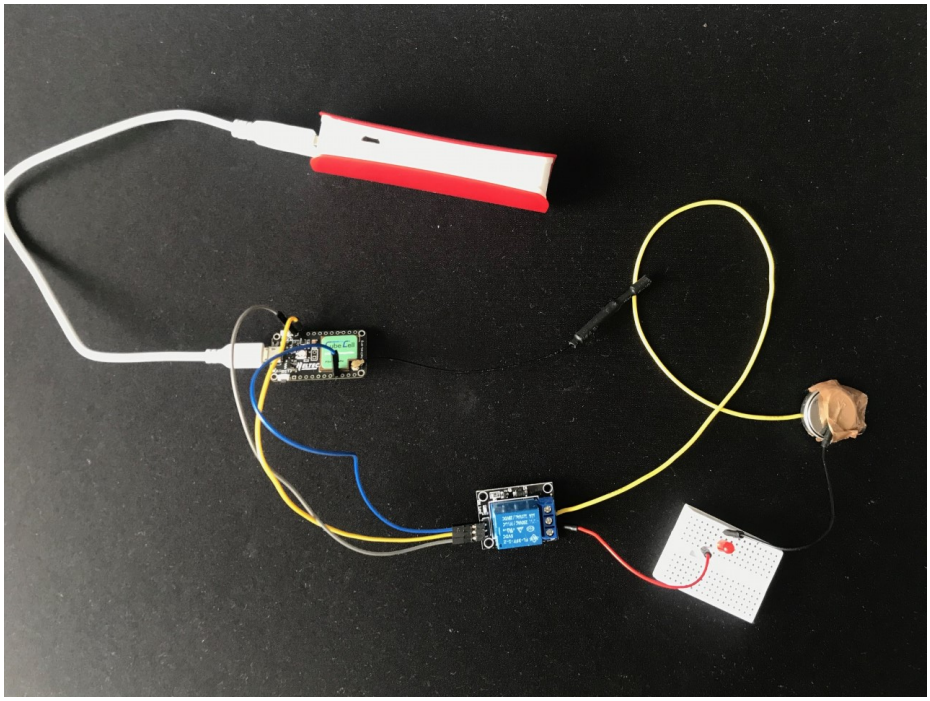


Figura 3.9: Led (alimentado con una pila) que emula el funcionamiento del pastor eléctrico conectado a una placa de desarrollo HTCC-AB01 por medio de un relé.

3.3.1 HTCC-ab01

Este microcontrolador de Heltec pertenece a la serie de productos Cubecell de bajo consumo (ver 3.2.1), diseñados para el desarrollo de dispositivos con comunicaciones LoRa.

3.3.2 KY-019

Este relé monocanal [61] es utilizado para controlar circuitos de AC o de distinto voltaje que el microcontrolador. El relé actúa como un interruptor que responde a la señal recibida del microcontrolador. Incorpora un led integrado que indica si la señal es alta o baja.

El KY-019 consiste en una resistencia de 1 MΩ, un led, un diodo rectificador y un relé de 5 VDC capaz de manejar hasta 250 VCA y 10 A.

3.4 Codificación y envío de datos mediante LoRaWAN

La programación de los nodos finales se realiza utilizando el Integrated Development Environment (IDE) de Arduino [62], que facilita la escritura de código y su carga en la placa. En

esta sección se explica la configuración del IDE para los diferentes dispositivos y se detallan los aspectos principales del código implementado para cada uno de ellos.

3.4.1 Instalación y configuración del entorno de programación

Una vez descargado el software de la página oficial de Arduino e instalado, para poder compilar y cargar el código en una placa, es necesario configurar el entorno de programación para esta.

Primero, conectar la placa al ordenador mediante el puerto USB y proceder a la configuración del la herramienta. Para ello, en el menú superior, acceder a "Archivo" > "Preferencias" y, en "Gestor de URLs adicionales de tarjeta" añadir `https://resource.heltec.cn/download/package_CubeCell_index.json`. Acto seguido, en el menú "Herramientas", acceder a "Gestor de tarjetas" e instalar la última versión de "CubeCell Development Framework".

A continuación, ir a "Herramientas" > "Puerto" y seleccionar el puerto que va a ser utilizado para la comunicación con la placa (ver figura 3.10).

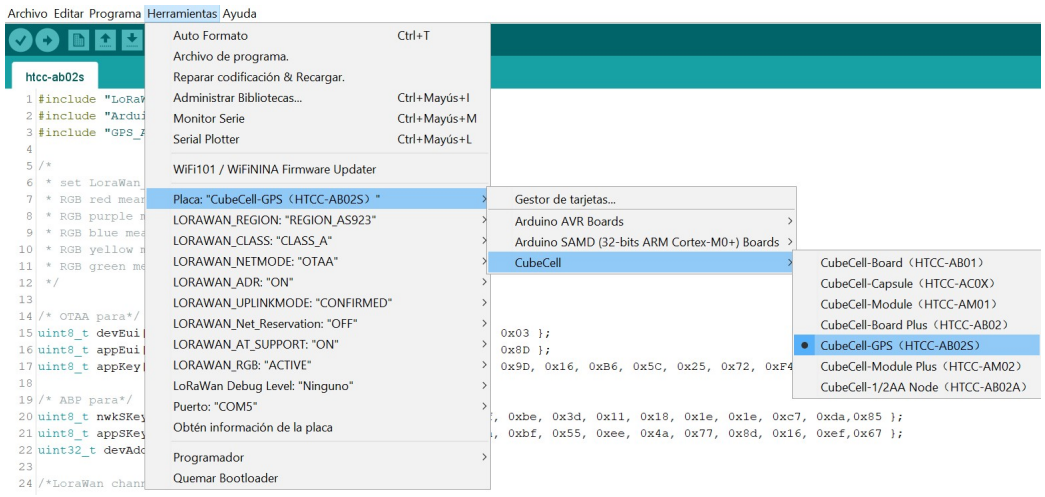


Figura 3.10: Selección del microcontrolador HTCC-AB02S en el IDE Arduino.

Una vez seleccionada la placa, configurar los parámetros para la comunicación vía LoRaWAN en el menú herramientas. La configuración queda como se indica en la figura 3.11.

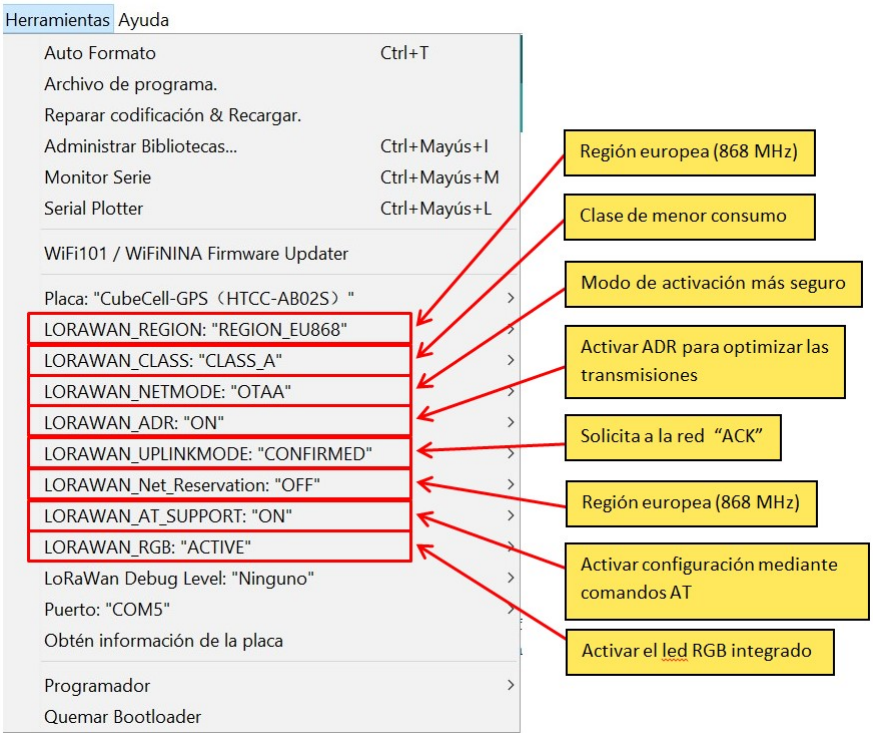


Figura 3.11: Configuración del IDE Arduino para el microcontrolador HTCC-AB02S con conexión LoRaWAN.

3.4.2 Codificación de datos

Para la transmisión de datos mediante LoRaWAN [63], tanto en mensajes de enlace ascendente como descendente, es importante reducir al máximo el tiempo en el aire del mensaje, consiguiendo así optimizar la duración de la batería y cumplir las restricciones de tiempo de transmisión máximo por dispositivo y día que establece TTN, que es siguiente:

- Una media de 30 segundos de tiempo en el aire en enlaces ascendentes.
- Un máximo de 10 mensajes de enlace descendentes (incluyendo mensajes de confirmación).

Para reducir el tiempo en el aire, a parte de otros parámetros, se debe optimizar la carga útil del mensaje. Para ello, es necesario codificar el mensaje [64] a nivel de bit, utilizando los desplazamientos y las máscaras.

A continuación, se detalla un fragmento del código implementado en el nodo final de localización GNSS HTCC-AB02S:

```
static void prepareTxFrame( uint8_t port )
{
    GNSSConection();

    double latitudeOrigin  = Air530.location.lat();
    double longitudeOrigin = Air530.location.lng();

    int latitudeInteger  = latitudeOrigin * 1000000;
    int longitudeInteger = longitudeOrigin * 1000000;

    appDataSize = 8;

    appData[0] = (byte)((longitudeInteger & 0xFF000000) >> 24);
    appData[1] = (byte)((longitudeInteger & 0x00FF0000) >> 16);
    appData[2] = (byte)((longitudeInteger & 0x0000FF00) >> 8);
    appData[3] = (byte)((longitudeInteger & 0x000000FF));

    appData[4] = (byte)((longitudeEntero & 0xFF000000) >> 24);
    appData[5] = (byte)((longitudeEntero & 0x00FF0000) >> 16);
    appData[6] = (byte)((longitudeEntero & 0x0000FF00) >> 8);
    appData[7] = (byte)(longitudeEntero & 0x000000FF);
}
```

Después de obtener las coordenadas de la ubicación, estas son convertidas a números enteros, multiplicando cada una por 1 000 000. Una vez obtenidos los enteros que representan a cada coordenada, se crea un array de 8 elementos, al que se añaden los bytes obtenidos después de utilizar la máscara perteneciente a cada uno de ellos y, posteriormente aplicar el desplazamiento correspondiente para eliminar los ceros. De esta forma, se obtiene el array que será enviado al servidor de IoT.

Para el resto de parámetros enviados en los distintos nodos finales, la codificación sigue el mismo método, intentando reducir al máximo la carga útil del mensaje enviado.

3.4.3 Envío de datos al servidor

Para realizar las tareas de conexión y envío de datos del nodo final a la puerta de enlace y viceversa, se implementa una máquina de estados. En el apéndice A.1 se muestra un fragmento de código perteneciente al nodo HTCC-ABS2. Este código está diseñado, especialmente, para realizar las pruebas de campo e imprimir la evolución de la conexión en su pequeña pantalla, lo que facilita enormemente estas tareas de prueba. A continuación, se detalla cada estado de la máquina:

- "Init". Se comprueba el estado del soporte por AT. Si está habilitado, se activa la recepción de esos comandos. Posteriormente se establecen los parámetros de la conexión.
- "Join". Se inicia el proceso de unión para conectarse a la red. Se muestra un diagrama de secuencia en la figura 3.12 [65], en el cual, el nodo final envía una petición de unión al servidor de red que, una vez aceptada la conexión, envía la "AppSKey" al servidor de aplicaciones.

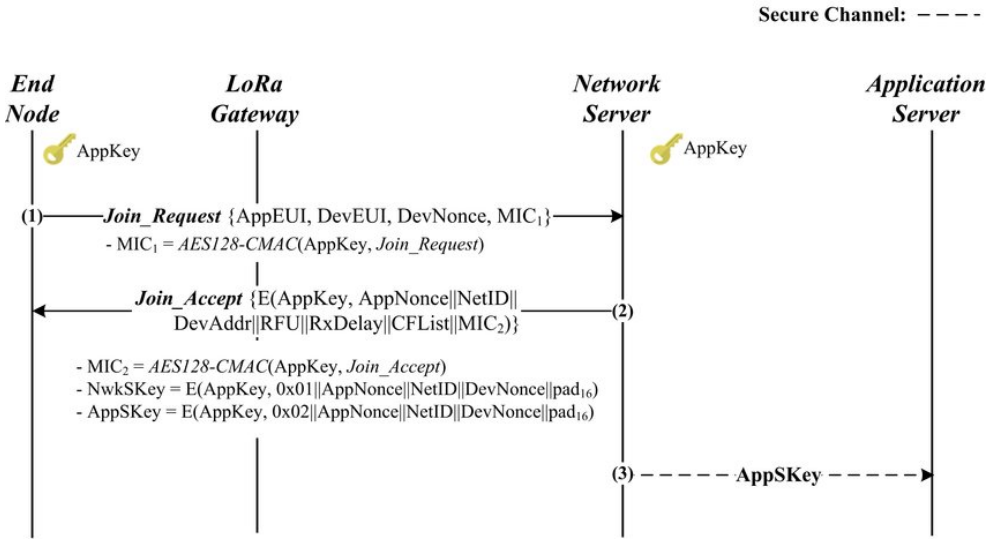


Figura 3.12: Procedimiento de "join" de un nodo final con el servidor de IoT mediante LoRaWAN.

- "Send". Obtiene los datos y los envía al servidor de red por el puerto indicado.
- "Cycle". Programa la siguiente transmisión de paquetes, sumando al ciclo predefinido un tiempo aleatorio.
- "Sleep". Recibe el "ACK" y se pasa al modo ahorro de energía hasta la siguiente transmisión.

Capítulo 4

Red de comunicaciones de internet de las cosas

Como red de comunicaciones de IoT, se utiliza The Things Network (TTN) [66], una red abierta global que ofrece un conjunto de herramientas para crear aplicaciones de IoT escalables y a bajo coste, con un sistema de cifrado robusto de extremo a extremo.

4.1 Introducción a The Things Network

TTN es una iniciativa basada en la comunidad para establecer una red global de IoT basada en la tecnología LoRaWAN. La iniciativa fue lanzada en 2015 y actualmente existen unas 18 000 pasarelas en funcionamiento en más de 150 países.

Una vez un usuario es miembro de TTN, puede agregar pasarelas a la comunidad y crear aplicaciones conectando sus propios dispositivos. Las pasarelas de la comunidad pueden ser utilizadas por cualquier dispositivo que pertenezca a la misma aunque esté registrada a nombre de otro usuario, por lo que si un nodo se conecta a una pasarela de TTN, los datos se procesarán hasta los servidores centrales, independientemente de que las pasarelas y el nodo sean del mismo propietario o no.

Cabe destacar la importancia que tiene la comunidad mundial que forman los miembros de TTN, posibilitando la participación en el foro oficial tanto para proporcionar ayuda como para recibirla.

Una vez creada una cuenta de usuario [67], se pueden registrar dispositivos, puertas de enlace y crear aplicaciones, posibilitando su gestión desde la propia consola y visualización de los paquetes que recibe tanto de enlace ascendente como descendente.

4.2 Registro de una puerta de enlace TTN

Una puerta de enlace es un dispositivo que hace de intermediario entre los nodos finales y un servidor de red (ver figura 4.1). Las puertas de enlace se conectan a los nodos mediante LoRaWAN y al servidor central por medio de conexiones Internet Protocol (IP) estándar. Es de tener en cuenta que los paquetes que envía un nodo final pueden ser recibidos por múltiples puertas de enlace, siendo el servidor central el encargado de la deduplicación de los mismos.

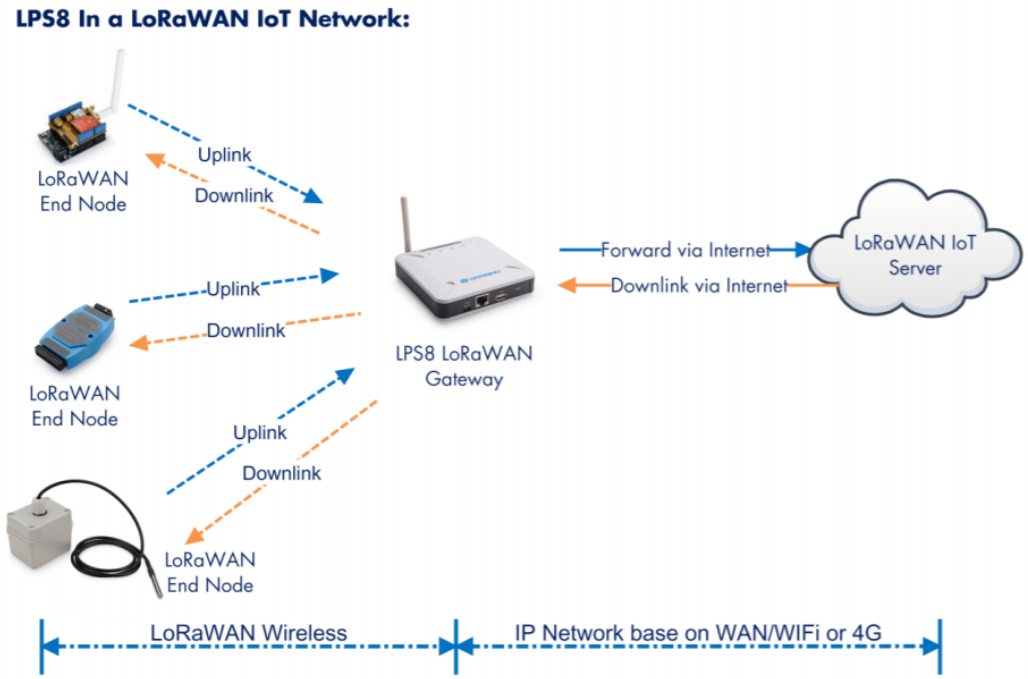


Figura 4.1: Representación del funcionamiento general de la puerta de enlace.

La puerta de enlace [68] adquirida para este proyecto es la Dragino LPS8 4.1. Este dispositivo es una solución económica, que utiliza un módulo Semtech compatible con LoRaWAN y proporciona 10 rutas de demodulación en paralelo. En cuanto a la conectividad a internet, permite la conexión por wifi o por Ethernet.

Al encender el dispositivo, se genera de forma automática una red wifi con la contraseña por defecto “dragino+dragino”. Después de conectarse a esta red, se accede a su configuración mediante la IP 10.130.1.1 introduciendo el usuario por defecto “root” y la contraseña “dragino”.

Una vez dentro del menú de configuración, en la barra superior, acceder a la pestaña “LoRa” y en “Radio Settings” ajustar el valor de ”Frequency Plan“ a la frecuencia europea, que son 868 Mhz.

Una vez ajustada la frecuencia, se entra en la pestaña “LoRaWAN”. En la configuración del servidor se selecciona el servidor europeo de TTN “TTN-router-EU” y se copia el identificador del dispositivo. Posteriormente este identificador será utilizado para registrar la pasarela en el servidor central.

Una vez realizados los pasos anteriores, ir a la pestaña “Home” y comprobar que todos los servicios funciona correctamente. La pantalla que se muestra, verificando que la configuración es correcta, es la que aparece en la figura 4.2.

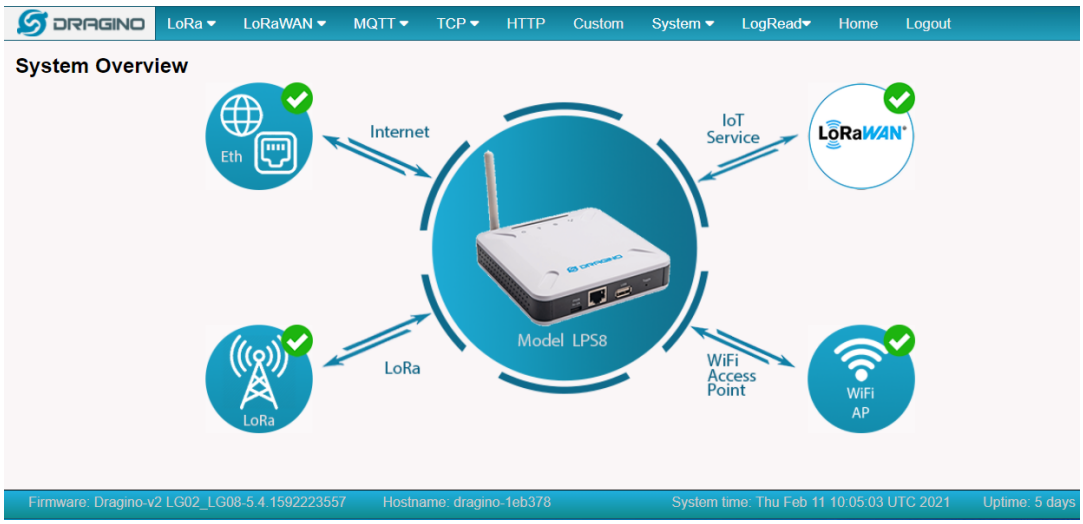


Figura 4.2: Pantalla principal de la configuración de la puerta de enlace donde analiza el estado de los servicios ofrecidos.

Una vez verificado que la puerta de enlace tiene los parámetros correctamente configurados, acceder a la consola de TTN, en la que se muestra una pantalla (ver figura 4.3) donde permite al usuario acceder a la sección de aplicaciones o a la de puertas de enlace.

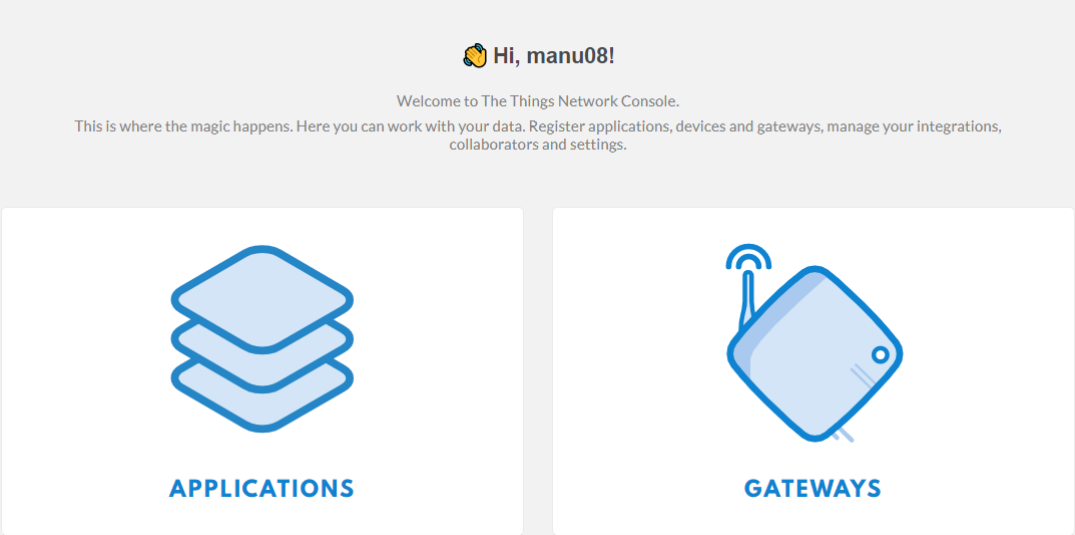


Figura 4.3: Pantalla principal de la consola de TTN.

Para el registro de la puerta de enlace, acceder a la sección de "gateways". Una vez dentro, pulsar en "register gateway" y cubrir el formulario que aparece, como se muestra en la figura 4.4.

Figura 4.4: Formulario de registro de una puerta de enlace en un servidor TTN

Al finalizar el proceso, se verifica que la puerta de enlace esté correctamente conectada, mostrando "status" como "connected".

4.3 Registro de dispositivos TTN

En esta sección se muestra el procedimiento de añadir un nodo final a la red TTN y el de procesar los datos desde la consola.

En el menú principal de la consola mostrado en la figura 4.3 se accede al apartado de "applications". Dentro de esta, se pulsa en el botón "add application". Para cubrir el formulario que aparece, se siguen las instrucciones indicadas en la figura 4.5.

Figura 4.5: Formulario de registro de una aplicación TTN

Para añadir un dispositivo a una aplicación dentro de TTN, son requeridos tres códigos. Estos códigos pueden estar proporcionados por el fabricante, en caso de obtener un producto terminado, o pueden ser asignados por el programador, en caso de ser nodos creados con placas de desarrollo. Los códigos son los que se detallan a continuación:

- DEV EUI. Identificador único para cada dispositivo.
- APP EUI. Código de identificador para cada aplicación definida en TTN.
- APP KEY. Clave para segurizar las comunicaciones con cada dispositivo.

Antes de añadir un dispositivo, acceder a la aplicación a la que se va a añadir y crear la "APP EUI" del dispositivo. Para ello, navegar hasta "add EUI" dentro de "Settings" > "EUIs".

Una vez la "APP EUI" ha sido añadida a la aplicación, se pordrá añadir el dispositivo como se indica en la figura 4.6.

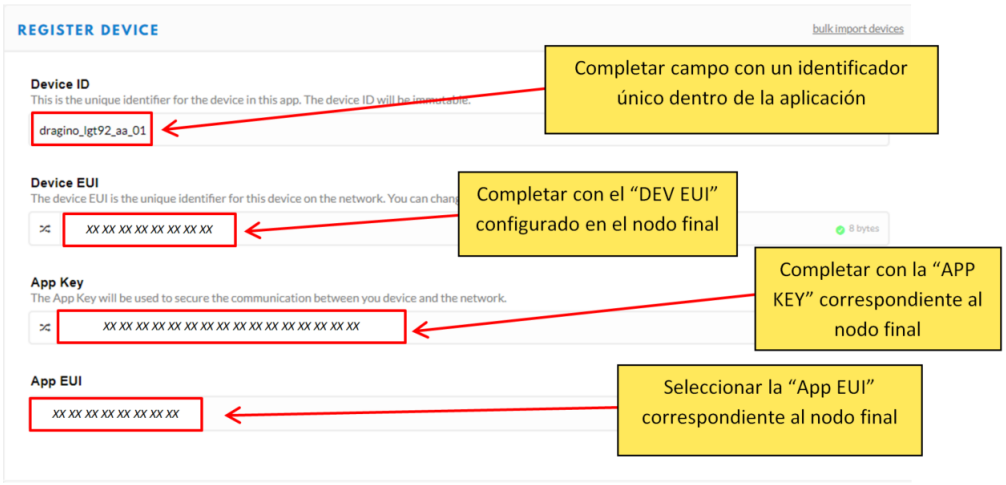


Figura 4.6: Formulario de registro de una dispositivo en una aplicación dentro de TTN

Al finalizar el proceso, se verifica que el dispositivo esté correctamente conectado, comprobando que su "status" como "connected".

4.4 Visualización de datos y estructura de la consola

En la consola de TTN, dentro del menú "Applications" existen 6 pestañas que, a continuación se detallan:

- Overview. Muestra los principales parámetros de la aplicación, el número de dispositivos que la forman y sus administradores.
- Devices. Aparece un listado con los dispositivos que componen la aplicación, marcando con un círculo de color verde los nodos activos.
- Payload Formats. Desde esta pestaña se permite implementar funciones en JavaScript para la decodificar, transformar y validar la carga útil recibida de los enlaces descendentes y devolver los campos obtenidos en formato JavaScript Object Notation (JSON). Desde este apartado también se permite codificar los mensajes recibidos, de enlace ascendente, en binario para ser enviados a los nodos finales.
- Integrations. Muestra las integraciones realizadas con aplicaciones externas.
- Data. Apartado en el cual se muestran los datos recibidos en el servidor. Estos datos puedes ser enviados desde cualquier dispositivo que forme parte de la aplicación seleccionada. Como se refleja en la figura 4.7, el mensaje aparece de dos formas, una es la carga útil del mensaje y la otra es el JSON generado por la propia aplicación.

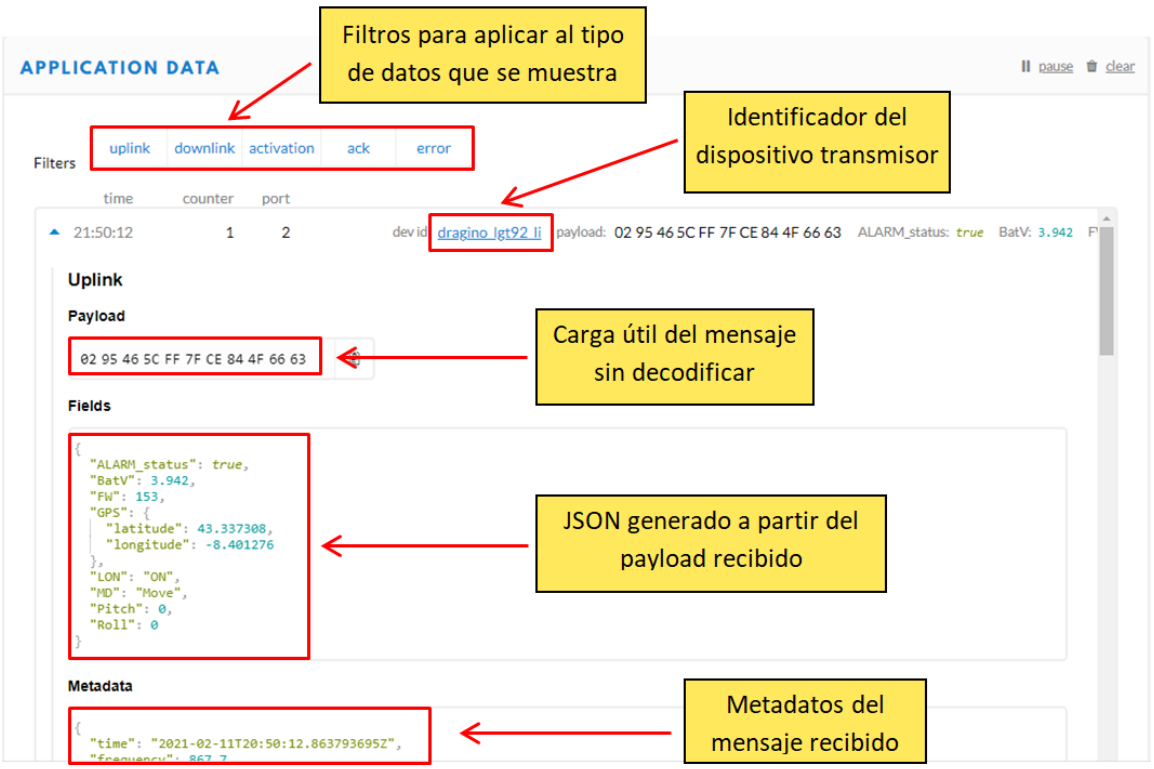


Figura 4.7: Desglose de los datos mostrados por la consola TTN dentro de una aplicación.

- Settings. Pestaña desde la cual se permite gestionar la aplicación TTN y modificar su configuración.

4.5 Procesamiento de datos

4.5.1 Decoder

Los datos recibidos en el servidor TTN están codificados tal y como se enviaron por los dispositivos. Es labor del desarrollador desfragmentar los datos recibidos, validarlos y procesarlos de forma adecuada. En las siguientes líneas se explica como hacerlo.

Dentro de ”Payload formats”, se encuentra la pestaña ”decoder”. En esta, de forma opcional, se puede implementar un método en JavaScript con dos parámetros de entrada (carga útil y puerto) que devuelve un resultado en formato JSON. Este script se ejecuta de forma automática para cada paquete que se recibe de cualquier dispositivo de la aplicación a la que pertenece. Se utiliza para transformar los datos que llegan de los de los nodos finales, en información en formato JSON. Para ello, puede ser necesario trabajar a nivel de bit, utilizando

desplazamientos y máscaras.

Siguiendo con el caso de uso explicado en la sección 3.4, a continuación, se expone el código de decodificación, que se ejecuta al recibir los datos enviados por cada dispositivo perteneciente a esta aplicación de TTN:

```
function Decoder(bytes, port) {

  return {
    GPS: {
      latitude:
        (bytes[0]<<24 | bytes[1]<<16 | bytes[2]<<8 | bytes[3]) /
    1000000,

      longitude:
        (bytes[4]<<24 | bytes[5]<<16 | bytes[6]<<8 | bytes[7]) /
    1000000
    }
  };
}
```

La carga útil de los datos recibidos se almacena en la variable ”bytes” y el puerto por la que se recibe, perteneciente a los metadatos, se guarda en la variable ”port”. A continuación, se procesan estos datos y se devuelve el resultado en un objeto en formato JSON, llamado ”GPS”, con dos campos numéricos: ”latitude” y ”longitude”. El resultado se devuelve en este formato debido a que es la nomenclatura esperada por el servidor de aplicación al recibir localizaciones globales.

Para la decodificación de los datos, en primer lugar, se identifica que bit pertenece a cada campo, teniendo en cuenta la codificación del dispositivo transmisor. A continuación, se aplican los desplazamientos necesarios para deshacer la operación de codificación realizada en los nodos finales. Una vez colocado cada bit en su posición, se realiza una operación lógica OR entre los bits pertenecientes a cada campo, obteniendo de esta forma los enteros correspondientes a cada coordenada.

Para terminar el proceso de decodificación, se divide cada entero por 1 000 000 (número que había multiplicado a la coordenada original antes de su codificación), obteniendo así las coordenadas correspondientes a cada posición.

Los datos almacenados en la variable ”port” pueden ser utilizados para distinguir los tipos

de datos que se reciben, permitiendo así tener distintos dispositivos en la misma aplicación. En este caso, no se ha utilizado porque se han implementado aplicaciones distintas.

4.5.2 Encoder

Dentro de la pestaña ”encoder”, de forma opcional, se puede implementar un método para codificar la información que se envía a un nodo final mediante un enlace descendente. Este script, igual que en ”decoder”, se ejecuta automáticamente para cada paquete que se envía a un dispositivo.

En este proyecto, para los dispositivos de geolocalización, no se ha implementado ninguna función para los comandos de enlace descendente, ya que no son necesarios. Para poder explicarlo, a continuación, se muestra el código perteneciente al control del pastor eléctrico.

```
function Encoder(object, port) {
  var bytes = [];

  if (object.pastor == "on"){
    bytes = [1];
  }
  if (object.pastor == "off"){
    bytes = [0];
  }

  return bytes;
}
```

El servidor TTN recibe, a través del protocolo MQTT, un objeto JSON con un ”on” o un ”off” y el ”encoder” lo codifica en un 1 o en un 0, devolviendo un array de bytes. Este array será enviado al nodo final para indicarle si debe encender o apagar el pastor eléctrico.

Capítulo 5

Servidor

En este capítulo se analizará el hardware sobre el que corre la aplicación servidora y se explicarán los distintos elementos software que forman la lógica de esta. El objetivo de este capítulo es servir de guía para que otra persona pueda replicar el proyecto utilizando dicho documento como manual.

El servidor se monta sobre un miniordenador de bajo coste, que puede ser ubicado en cualquier sitio con conexión a internet. Esto se podría hacer en otro tipo de servidores, en particular, en algún hosting con máquinas virtuales.

5.1 Hardware

Para el desarrollo de este proyecto, se utiliza un servidor físico. El hardware utilizado es una Raspberry Pi 4 Model B [31] con una tarjeta microSD de 32 GB donde se almacena toda la información del servidor, incluido el sistema operativo.

Como se refleja en la figura 5.1, la Rasberry Pi es un miniordenador de un tamaño muy reducido. Este dispositivo consta de una placa base sobre la que se monta un procesador, un chip gráfico, memoria RAM, etc. En la tabla 5.1 se muestran sus características principales.

Raspberry Pi 4 Model B	
Procesador	Broadcom BCM271 4 núcleos a 1.5 GHz
Memoria	2 GB SDRAM
Conectividad inalámbrica	IEE 802.11ac , Bluetooth 5.0, BLE
Conectividad cable	Gigabit Ethernet
Puertos USB 3.0	1
Puertos USB 2.0	1
Tipo tarjeta	MicroSD
Corriente funcionamiento	5 VCC a 3 A
Temperatura funcionamiento	$0 \sim +50$℃
Dimensiones	85 mm x 53 mm

Tabla 5.1: Características principales de la Raspberry Pi 4 Model B.

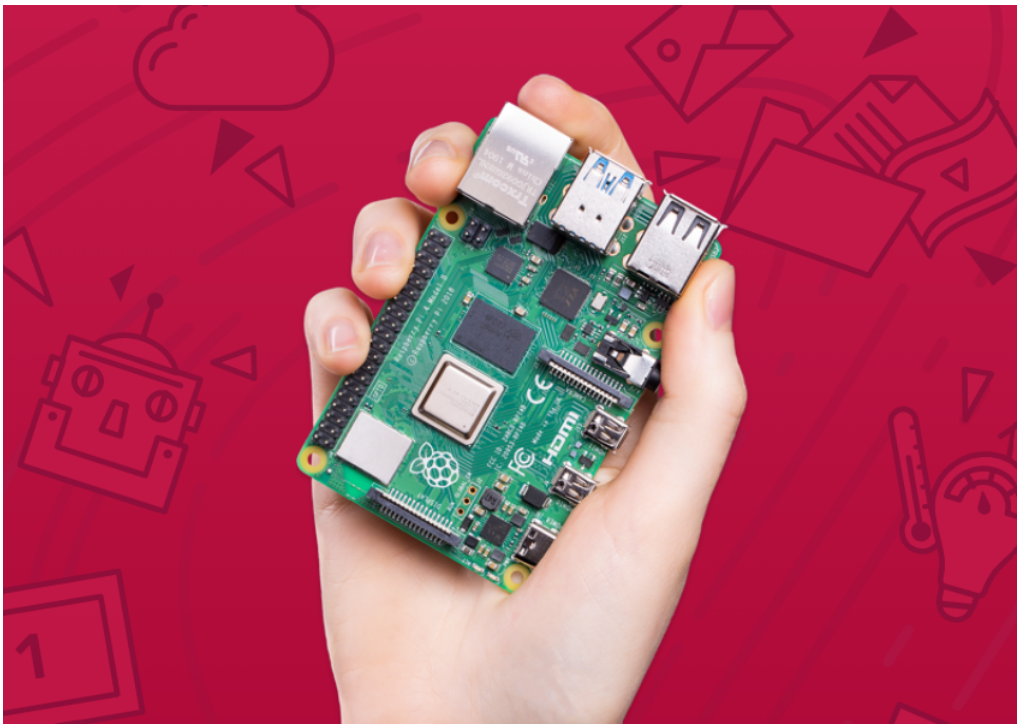


Figura 5.1: Imagen de una Raspberry Pi 4 sobre la palma de una mano.

5.2 Software

Se ha instalado Home Assistant (HA) como software central del servidor. HA es un software de automatización de dispositivos gratuito y de código abierto. Tanto este sistema como sus extensiones, están escritos en Python, siendo su enfoque principal el control local y la privacidad.

Dentro este software, basado en el SO Buildroot LTS, se instalarán el resto de aplicaciones utilizadas para hacer posible el funcionamiento del sistema. En este, se alojará la herramienta de desarrollo para implementar la lógica de negocio, la gestión de la base de datos, controles avanzados del servidor, interfaz gráfica, etc.

5.2.1 Instalación de Home Assistant

Para instalar HA en la Raspberry Pi 4 Model B se han seguido los pasos que se detallan a continuación:

- Descargar la imagen del software adecuada para el hardware en el que va a ser instalado. En este caso se ha descargado la versión de 64 bits de GitHub [69].

- Descargar "balenaEtcher". Software de código abierto que será utilizado para montar la carpeta comprimida, descargada en el punto anterior, en una tarjeta microSD creando así una "Live SD" [70].

- Utilizar balenaEtcher para montar el archivo descargado en la tarjeta microSD.

- Una vez montado el archivo en la tarjeta de memoria, insertarla en la Raspberry Pi y conectarla a la fuente de alimentación y al router.

- Una vez encendido el miniordenador, este se iniciará, se conectará a internet e instalará de forma automática la última versión de HA.

5.2.2 Conexión al servidor

Una vez el software está montado en la Raspberry, es necesario configurarlo. Para ello, hay que conectarse a esta a través del puerto 8123, desde un navegador web. Después de unos minutos, se termina de instalar la última versión del software y aparece una pantalla como la mostrada en la figura 5.2, que permite crear una cuenta de usuario. Una vez creada la cuenta, se tendrá acceso al panel de control por defecto de HA.

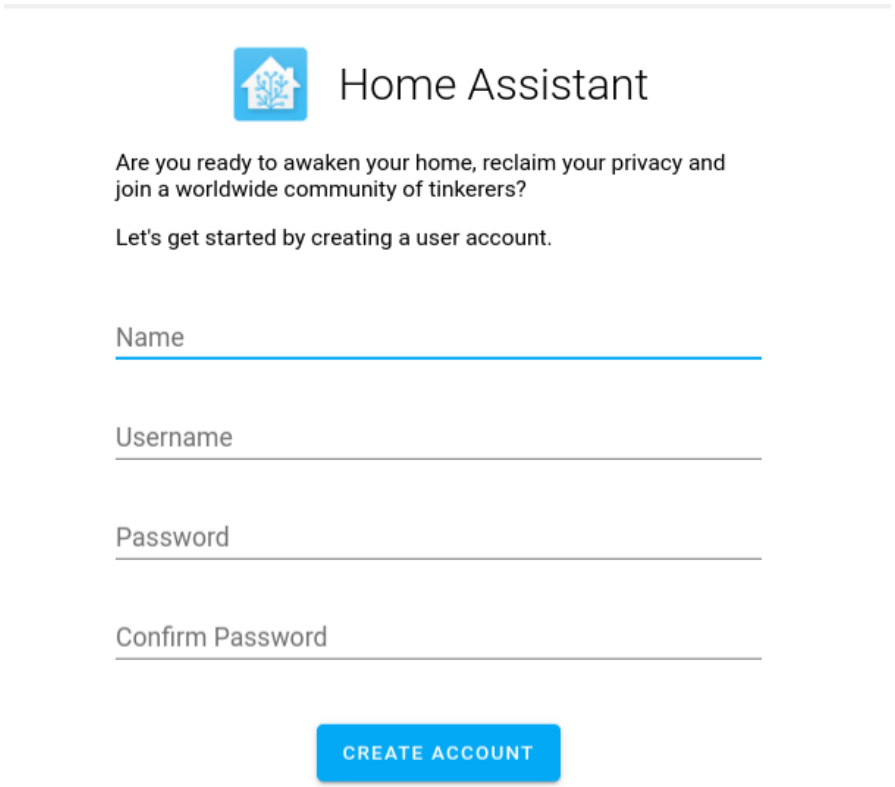


Figura 5.2: Pantalla mostrada por HA que permite crear una cuenta de usuario.

5.2.3 Configuracion inicial

A continuación, se establecen parámetros básicos de la creación de la cuenta, indicando nombre del sitio que se desea automatizar y su ubicación. También es en este punto donde se desbloquean las configuraciones avanzadas de usuario.

El próximo paso es configurar los add-ons que se necesitan. Estos son aplicaciones adicionales que permiten al usuario ampliar las funcionalidades de HA. A través de estos complementos, se pueden integrar con facilidad aplicaciones en el propio servidor y, posteriormente, utilizarlos de forma similar a las funcionalidades nativas del mismo SO.

Inicialmente, se instalan los complementos e integraciones básicas, para obtener un mayor control del sistema y poder establecer comunicación con los nodos finales configurados en el capítulo 3.

5.2.4 Instalación de complementos e integraciones

File Editor

Es un editor de texto que se utilizará para modificar archivos del sistema. Eso permitirá añadir configuraciones e integraciones que no se pueden hacer desde la interfaz gráfica.

Para instalar este complemento es necesario acceder en la barra lateral izquierda a "Supervisor" > "Add-on store", y aquí buscar "File Editor" e instalarlo. Una vez instalado, lo iniciamos y marcamos las opciones "Start on boot" y "Show in sidebar", consiguiendo así tener un rápido acceso a este recurso desde el panel lateral y que se inicie en cada arranque del sistema.

Mosquitto broker

Mosquitto es un broker MQTT de código abierto. Este protocolo de comunicaciones será utilizado, por medio de este broker, para el intercambio de mensajes de HA con el servidor TTN y para el intercambio de mensajes entre distintos elementos del servidor de aplicaciones local.

Para instalarlo, buscar "Mosquitto broker" en "Add-on store".

Terminal y SSH

Este complemento, incluye una herramienta de línea de comandos para acceder a la API de HA. También permite iniciar sesión de forma remota usando Secure Shell (SSH).

Para instalar esta herramienta, buscar "Termina & SSH" en "Add-on store".

Node-RED

Herramienta de desarrollo fundamentada en flujo para programación visual, utilizada para gran parte de la lógica del backend de este proyecto.

Esta herramienta permite el diseño de flujos de datos a través de distintos componentes de un sistema más grande, permitiendo añadir nodos existentes o desarrollar nodos nuevos y conectarlos entre sí con el fin de que se comuniquen entre ellos.

Samba

Implementación libre del protocolo de archivos compartidos de Windows para sistemas de tipo UNIX. Con el uso de esta herramienta podemos acceder desde Windows a los directorios de HA almacenados en la Raspberry Pi.

MQTT

MQTT es un protocolo de comunicaciones, compuesto por publicadores y suscriptores, que utilizan un ancho de banda de red mínimo. En este caso de estudio, se utiliza para establecer comunicación del servidor local con el servidor de la red TTN y dentro del propio HA.

Una vez instalado 5.2.3, se debe integrar al servidor. Para ello se debe acceder en el panel de la izquierda a "Configuration" > "Integrations" > "Add integration" y buscar MQTT.

Base de datos

HA utiliza una base de datos para almacenar eventos y parámetros para el historial y el seguimiento. La base de datos predeterminada utilizada es SQLite [46] y el archivo de la base de datos se almacena en su directorio de configuración, sin embargo, se pueden utilizar otras bases de datos.

5.2.5 HACS

Home Assistant Community Store (HACS) es una integración que brinda una poderosa interfaz de usuario que permite manejar descargas de integraciones y complementos personalizados.

Para la instalación de HACS se utiliza el complemento SSH. Se ejecutará el script de instalación proporcionado en el sitio web oficial como se muestra en la figura 5.3 [71].

```
~ $ wget -q -O - https://hacs.xyz/install | bash -
INFO: Trying to find the correct directory...
INFO: Found Home Assistant configuration directory at '/config'
INFO: Changing to the custom_components directory...
INFO: Downloading HACS
Connecting to github.com (140.82.121.3:443)
Connecting to github.com (140.82.121.3:443)
Connecting to github-releases.githubusercontent.com (185.199.110.154:443)
saving to 'hacs.zip'
hacs.zip             100% |*************************************************
'hacs.zip' saved
INFO: Creating HACS directory...
INFO: Unpacking HACS...
INFO: Removing HACS zip file...
INFO: Installation complete.

INFO: Remember to restart Home Assistant before you configure it
~ $
```

Figura 5.3: Ejecución del script de instalación de HACS.

Una vez instalado y reiniciado el servidor, acceder a “HACS”, dentro del menú ”Integrations“ y aceptar el uso de todos sus componentes.

El siguiente paso es registrar el dispositivo. HACS usa un flujo Open Authorization (OAuth) para la autenticación de los dispositivos contra la API de GitHUb. Copiar el código que aparece en la parte inferior (ver figura 5.4) y hacer click en el enlace que aparece a GitHub.

Se abre una pantalla donde se debe insertar el código copiado en el paso anterior y hacer click en “Authorize hacs”. A continuación, aparece una imagen como la mostrada en la figura 5.5.

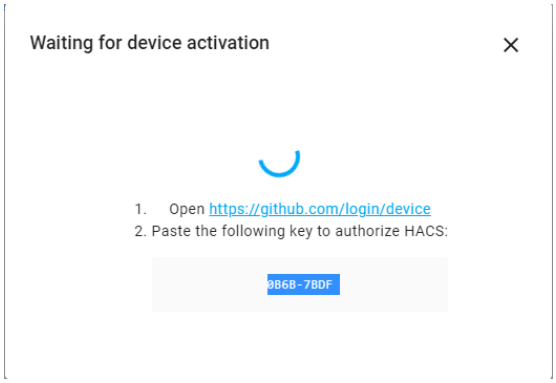


Figura 5.4: Autenticación del dispositivo contra la API de GitHub.

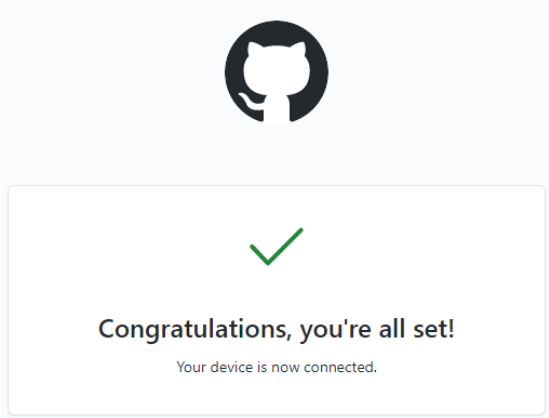


Figura 5.5: Confirmación de conexión.

Acto seguido, volver a HA y hacer click en "Finish", que completará la configuración de HACS y aparecerá visible en la barra lateral.

Administración de cuentas

Para poder crear distintos usuarios con accesos restringidos, ir a "Configuration" > "Users" y pulsar sobre "Add user". Introducir el nombre y la contraseña del nuevo usuario e indicar si se le conceden permisos de administrador.

Posteriormente, permitir o denegar el acceso a cada pestaña del panel principal, accediendo a "Edit View" > "Visibility".

Snapshots

Permitir realizar copias de seguridad periódicas frecuentes y restaurar todos los datos de la instancia de HA es una tarea muy importante para la recuperación de errores del sistema. Si surge cualquier problema, se podrá restaurar desde una "snapshot" reciente en cualquier dispositivo. A continuación, se configurará, para que de forma automática, las copias sean realizadas de forma simultánea en el propio servidor y en Google Drive.

Para ello, se deben de seguir los siguientes pasos:

- Añadir el repositorio de GitHub correspondiente [72]. Para ello, acceder a "Supervisor" > "Add-on Store" > "Repositories" e introducir la URL en el cuadro inferior.

- Instalar el add-on "Home Assistant Google Drive Backup" y marcar "Start on boot".

- Configuración. En la sección de configuración se pueden ajustar multitud de parámetros. En este proyecto se ha configurado para que la copia de seguridad se realice diariamente a las 1:00 h, que es cuando se supone que el servidor tendrá una carga baja de trabajo. Se ha configurado para que se almacenen las 4 últimas copias tanto en la nube como en el propio servidor. Se muestra el archivo de configuración en la Figura 5.6.

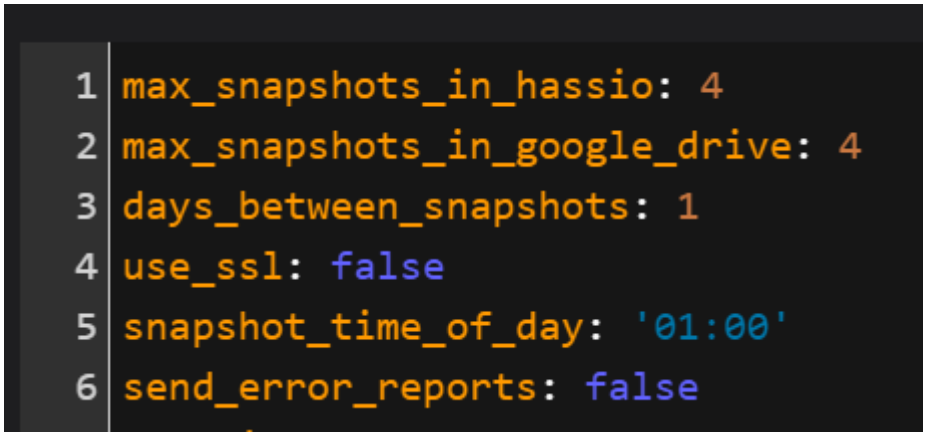


Figura 5.6: Configuración del "Home Assistant Drive Backup".

- Autenticación. Para completar la creación de backups en la nube faltaría autenticar el dispositivo con Google. Para ello, hacer click en "Open web UI", pulsar sobre "Authenticate with Google" y copiar el código de autorización que aparece, en HA.

Para que los complementos e integraciones instalados funcionen, reiniciar el servidor HA después de completar la instalación de cada uno de ellos. Para hacerlo, ir a "Configuration" > "Server Controls" > "Restart".

Antes de reiniciar el servidor, es recomendable hacer click en "Check Configuration" para verificar que la configuración es válida y evitar errores en el reinicio del sistema.

5.3 Acceso externo al servidor

Para poder hacer posible el acceso desde el exterior de la red al servidor de la aplicación, se utilizará Duck DNS, que proporciona un servicio de Dynamic Domain Name System (DDNS) de forma gratuita. Además, se necesita configurar la red para que permita dicha conexión.

5.3.1 Utilización de Duck DNS

Para la configuración de Duck DNS [73] se seguirán los siguientes pasos:

- Se crea una cuenta en la página oficial. Una vez creada la cuenta, se genera un token, que debe ser copiado para utilizar posteriormente y se elige un subdominio. En este caso, se ha elegido `mcouto.duckdns.org`.

- Abrir HA e instalar el Add-on de de Duck DNS siguiendo los mismos pasos que el resto complementos instalados en la sección 5.2.4.

  En la configuración, introducir el dominio de Duck DNS y el token generado en el paso anterior. Además, se debe de cambiar a "true" el valor de la variable "accept_terms" de forma que la configuración quede, finalmente, como se muestra en la figura 5.7.

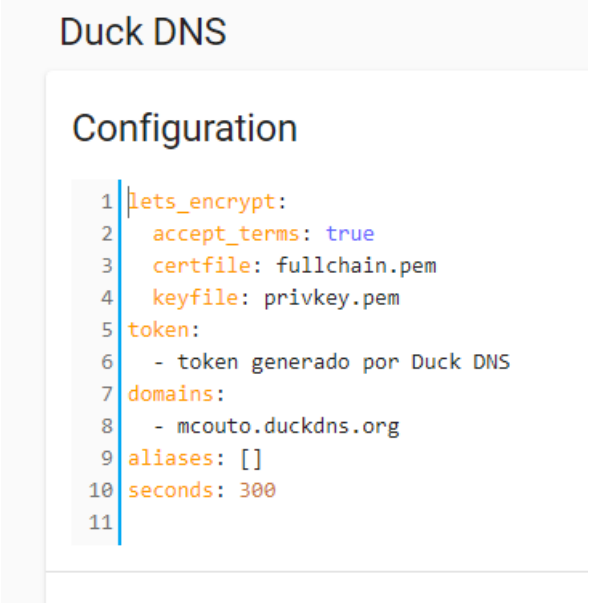


Figura 5.7: Archivo de configuración del add-on de Duck DNS en HA.

- Utilizando el "Editor File" editar el archivo "configuration.yalm" y añadir las líneas que aparecen en la Figura 5.8, donde se indica que se utilizará un certificado autocifrado y la url del dominio de DuckDNS con la que se accede al servidor.

```
# DuckDNS configuration for access to server from outdoor
http:
  ssl_certificate: /ssl/fullchain.pem
  ssl_key: /ssl/privkey.pem
  base_url: https://mcouto.duckdns.org:8123
```

Figura 5.8: Configuración de Duck DNS en el archivo configuration.yalm

- Para que surtan efecto los cambios realizados, reiniciar el servidor.

5.3.2 Configuración de red

Una vez instalado y configurado Duck DNS en HA, falta configurar la red local. Para esto, es necesario asignar una IP privada fija a cada elemento del sistema al que se desea tener acceso desde el exterior de la red local.

En este caso, se necesita tener acceso a la Raspberry Pi desde el exterior, por lo que hay que asignar una IP a este dispositivo. Posteriormente, será necesario realizar una redirección de puertos.

Cada router puede tener un menú de configuración distinto. En este caso, se utiliza un router ZTE F680 [74].

Para configurar una IP privada como estática, como se muestra en la figura 5.9, es necesario, una vez dentro del router, acceder en el menú de la izquierda a "LAN" > "DHCP Binding" y aquí enlazar la Media Access Control (MAC) del servidor con la IP fija que se desee. En este caso se ha asignado al servidor la IP 192.168.1.133.

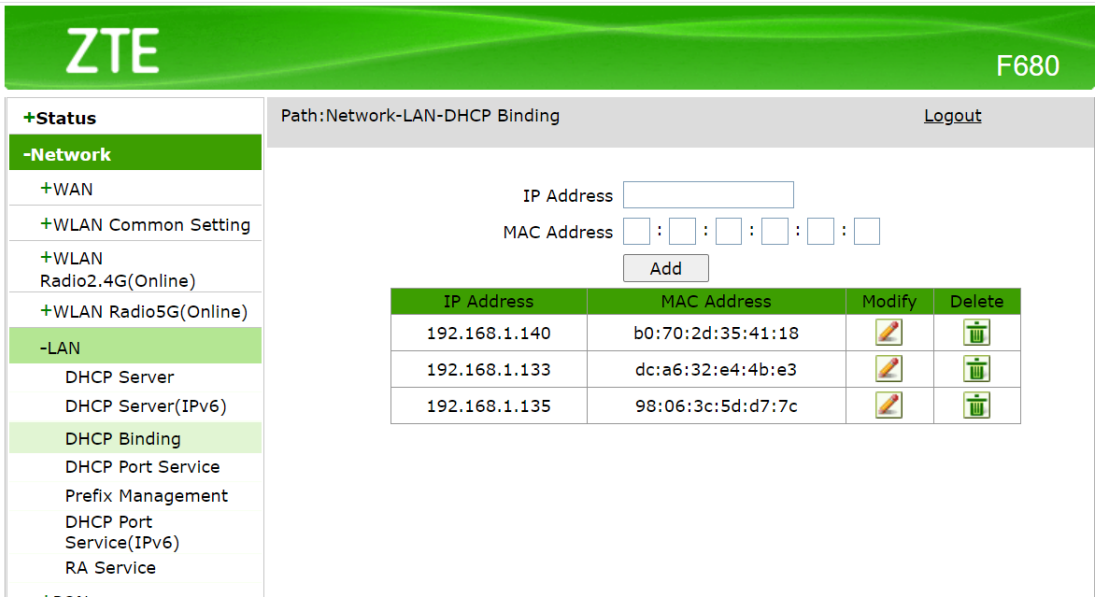


Figura 5.9: Configuración del router ZTE F680 donde se muestran distintas IPs privadas estáticas.

Para la configurar la redirección de puertos de acceso al servidor, será necesario acceder al router y, en el menú lateral izquierdo, acceder a "APPLICATION" > "PORT FORWARDING", y configurarlo como se muestra en la figura 5.10).

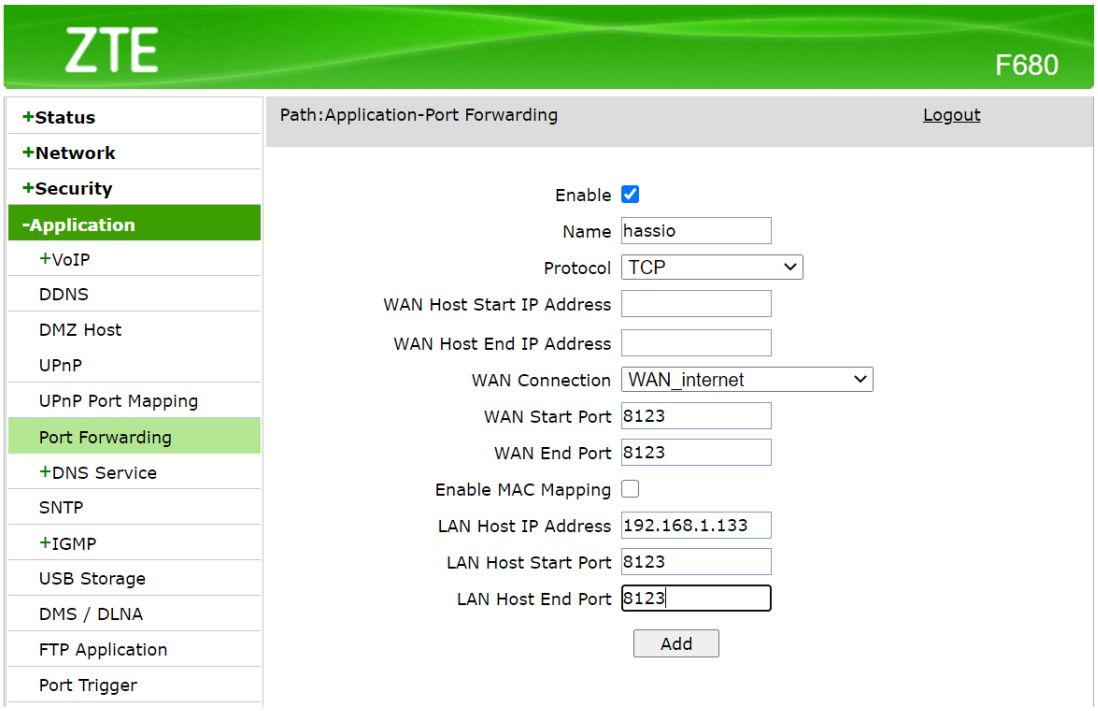


Figura 5.10: Configuración de la redirección de puertos en un router ZTE F680.

5.4 Implementación de la lógica de negocio

La parte central de la lógica de negocio de este proyecto es implementada utilizando Node-RED, herramienta de programación visual basada en flujos explicada en la sección 5.2.4.

En la figura 5.11 se muestra la estructura general de esta herramienta de programación visual dividida en las siguientes secciones:

- Panel izquierdo. En la parte izquierda de la pantalla, se localiza un panel donde se pueden encontrar los nodos preexistentes en la herramienta para se utilizados.
- Pestañas superiores. En la parte superior de la pantalla, aparecen las pestañas en las que se puede dividir el editor, permitiendo de esta forma una mejor organización del código.
- Panel derecho. En la parte derecha están los paneles de información, menús de configuración y la consola de "debug".
- Sección central. En la sección central del editor es donde se pueden conectar los nodos creando flujos.

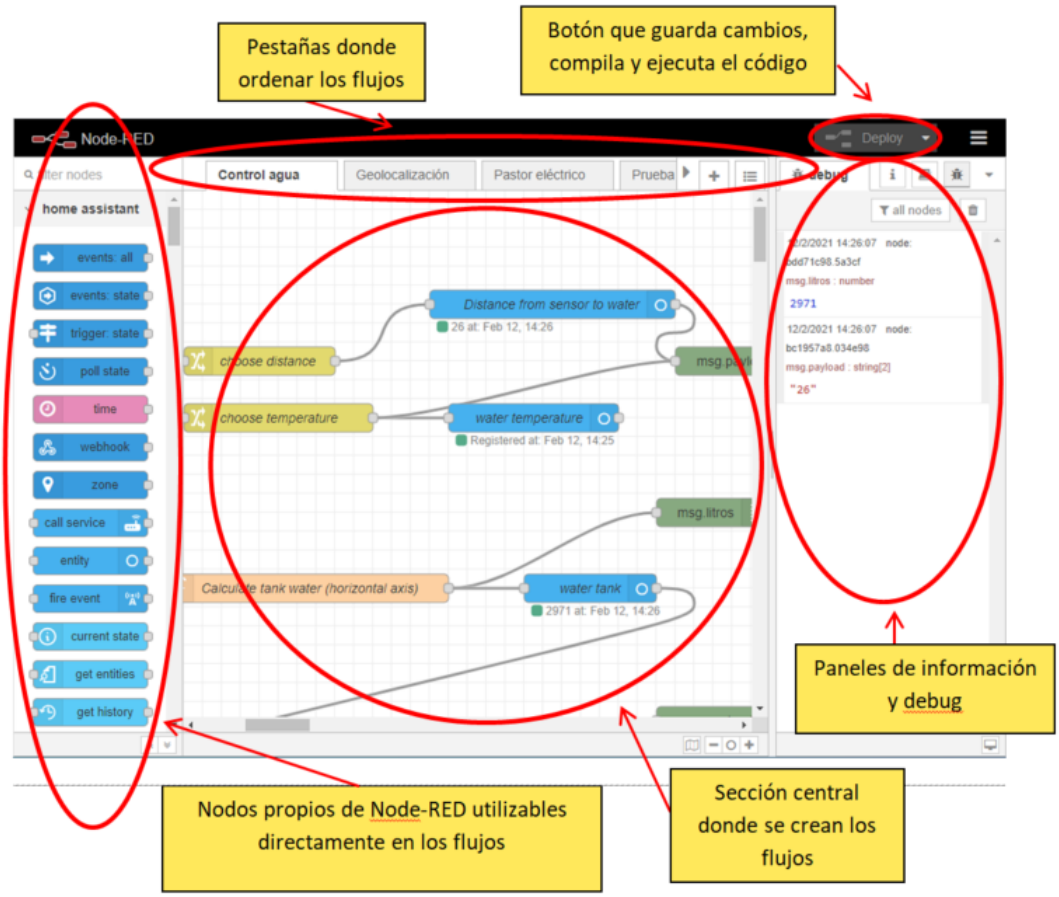


Figura 5.11: Estructura de la herramienta visual Node-RED.

5.4.1 Extracción de datos de TTN

Los servidores de TTN no almacenan los datos, lo que significa, que se necesita un método de extracción de datos en el momento que son recibidos por el servidor o, en caso contrario, se perderán. Para ello, existen dos posibilidades: mediante integraciones de aplicaciones o mediante MQTT.

En este proyecto, después de valorar las dos opciones, se ha decidido optar por la obtención de datos utilizando el protocolo MQTT, evitando así depender de otros servidores externos que, en un futuro podrían no ofrecer sus servicios o cambiar sus condiciones de uso, y no ajustarse a las especificaciones del proyecto.

El primer paso para la obtención de datos en Node-RED, mediante el protocolo MQTT, es añadir un nodo de tipo “mqtt in” al editor de flujo. Este nodo sirve para conectar el servidor a un broker MQTT y posibilitar la suscripción a un “topic” especifico. Al hacer doble click

sobre el nodo añadido, se despliega el menú de edición de dicho nodo, como se muestra en la Figura 5.12.

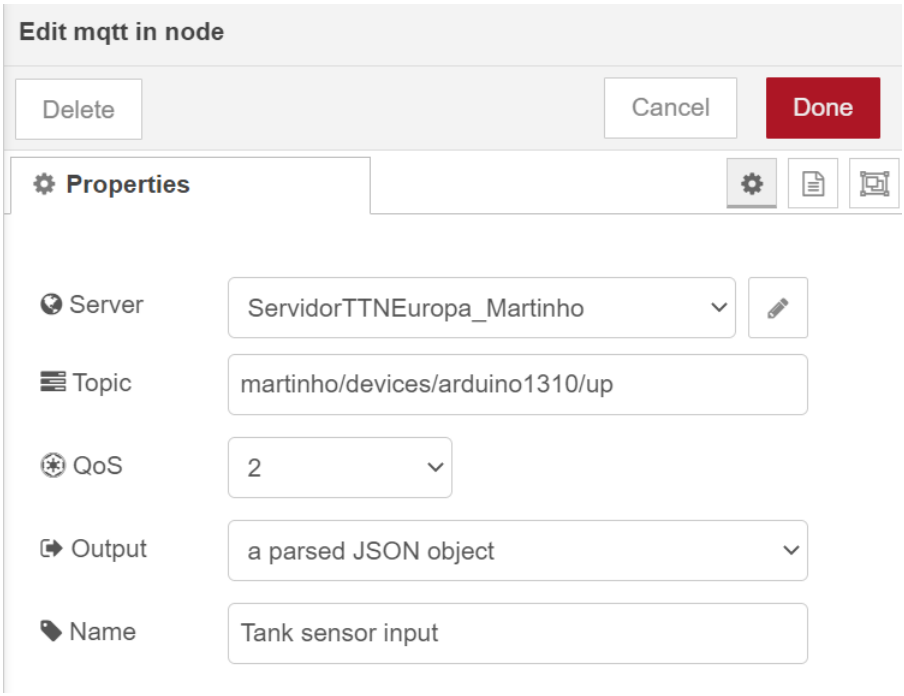
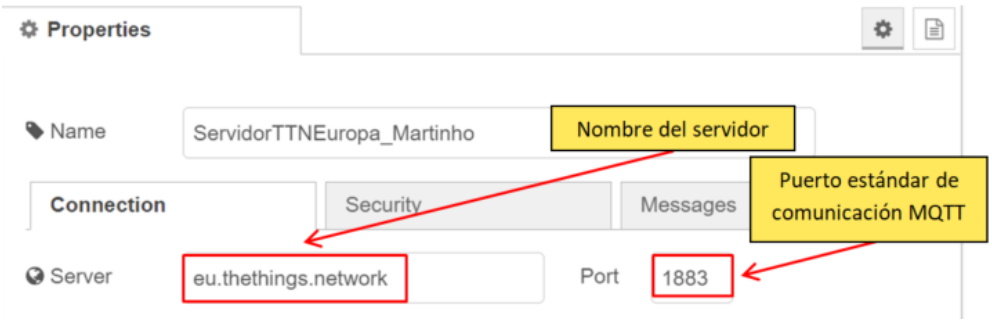


Figura 5.12: Menú de edición de un nodo ´´mqtt in" en Node-RED.

El "topic" al que suscribirse es <AppID>/devices/<DeviceID>/up. Siendo <AppID> el identificador de la aplicación en TTN y <DeviceID> el identificador del dispositivo.

Como formato de datos de salida introducir "a parsed JSON object", para indicar a la herramienta que se necesita generar un texto en formato JSON en la salida del nodo. En "Server" añadir un nuevo broker MQTT. Para ello, seleccionar la opción "Add new mqtt-broker". Al pulsar sobre el icono de la derecha se abre un nuevo formulario, como el explicado en la figura 5.13, que permite crear una nueva configuración de acceso a un broker.

Conectados a la salida de los nodos anteriores, van los nodos de procesamiento, que consumen el JSON que éstos proporcionan. Estos nodos pueden tener una funcionalidad predefinida o pueden ser nodos implementados en JavaScript por el desarrollador.



(a) Configuración de conexión.

(b) Configuración de seguridad.

Figura 5.13: Menú de configuración de un broker MQTT en Node-RED.

5.4.2 Creación de entidades

Una de las partes más interesantes de comunicación de Node-RED con HA es la creación de entidades. Para poder crear entidades en HA, es necesario seguir los pasos que se indican a continuación:

- Dentro de HA.
  - Crear en el directorio “custom_components” en la ubicación donde se encuentra el fichero “configuration.yalm”.
  - Dentro de “custom_components” crear el directorio “nodered”.
- En Node-RED.

- Instalar un componente complementario que incluye los nodos de conexión. Para ello, acceder a "palette", buscar "node-reed-contrib-home-assistant-websocket" e instalarlo.

- Configurar el servidor. Cada vez que se utiliza uno de los nodos añadidos en el paso anterior, es obligatorio indicarle el servidor donde se encuentra el HA donde se añadirá la entidad. Para configurar el servidor, seleccionar "Add new server". Una vez dentro, marcar la opción "I use the Home Assistant Add-on" y pulsar "Add". De esta forma, Node-RED añadirá las entidades en el HA donde se está ejecutando.

5.4.3 Implementación de funcionalidades

Una vez obtenidos los datos de los sensores, es necesario procesarlos. Para ello, es necesario utilizar el editor, conectar nodos entre sí, configurarlos y programarlos.

En la figura 5.14 se muestra el flujo de Node-RED, creado para recibir los datos de los localizadores GNSS desde el servidor TTN, mediante el protocolo MQTT, y dependiendo de sus valores, crear alertas.

Además, para la implementación y realización de pruebas, en la zona de la izquierda del flujo, se ha insertado un nodo llamado "simulator", para inyectar datos simulando la recepción de información de un sensor. En la zona de la derecha, además de los nodos que envían las alertas, se han añadido nodos de "debug", que muestran información en la consola de Node-RED facilitando así la depuración de errores.

En este flujo, a partir de los datos recibidos del dispositivo localizador, se comprueba si la posición es válida, para actualizar la ubicación en el mapa. De forma concurrente, se examina si se ha pulsado el botón de alarma y se analiza si el nivel de batería es bajo. Si alguna de las condiciones es cierta, se genera su respectivo mensaje de alerta y se envía vía WhatsApp®.

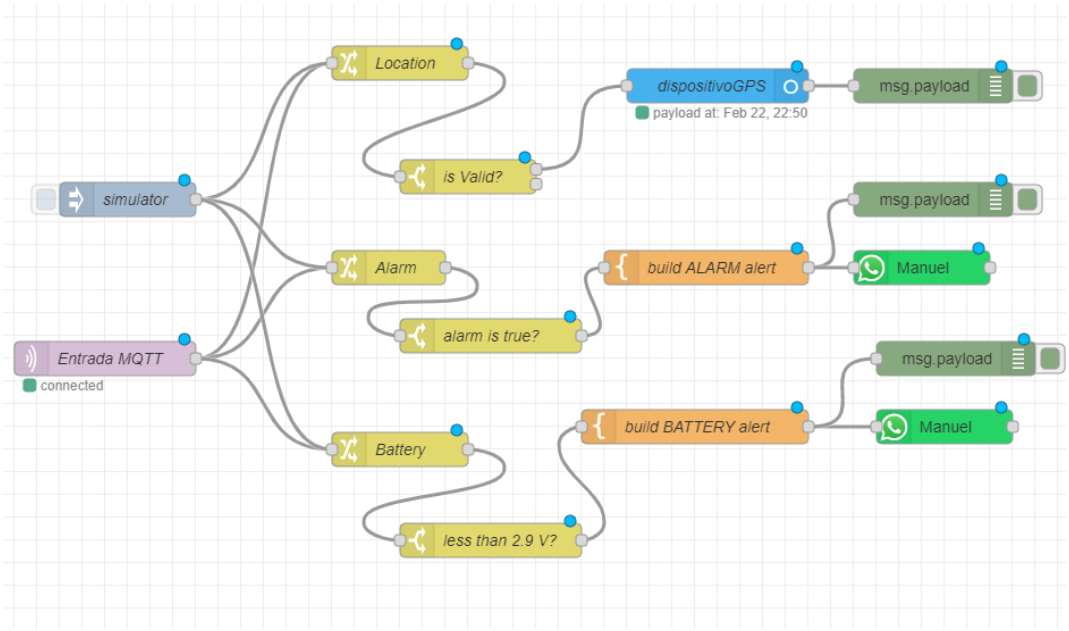


Figura 5.14: Flujo de Node-RED que procesa la información y genera alertas vía WhatsApp®.

5.5 Interfaz de usuario

La interfaz de usuario utilizada es la proporcionada por HA, denominada “Lovelace”. Es una interfaz amigable, intuitiva y altamente configurable.

Se utilizan tarjetas para mostrar los dispositivos, que pueden ser reales o virtuales. De forma nativa, son proporcionadas distintas tarjetas y, a través de HACS se pueden obtener otras de múltiples estilos.

Para añadir una tarjeta, acceder al panel principal y hacer click en los tres puntos situados en la esquina superior derecha. Al desplegarse el menú, pulsar en “Edit Dashboard”. En este panel se pueden añadir cartas y clasificarlas en distintas pestañas.

Para este proyecto se crearon las cartas que, a continuación, se indica:

- Una carta de tipo mapa, donde se muestra la ubicación de los animales en tiempo real y sus trayectorias. El tiempo de la trayectoria que se desea mostrar es configurable, lo que permite cubrir dos necesidades importantes:
  - La posición del ganado con sus trayectorias recientes. Como se muestra en la ver figura 5.15 esto permitirá al ganadero saber donde se encuentra cada animal en cada momento y su recorrido diario.

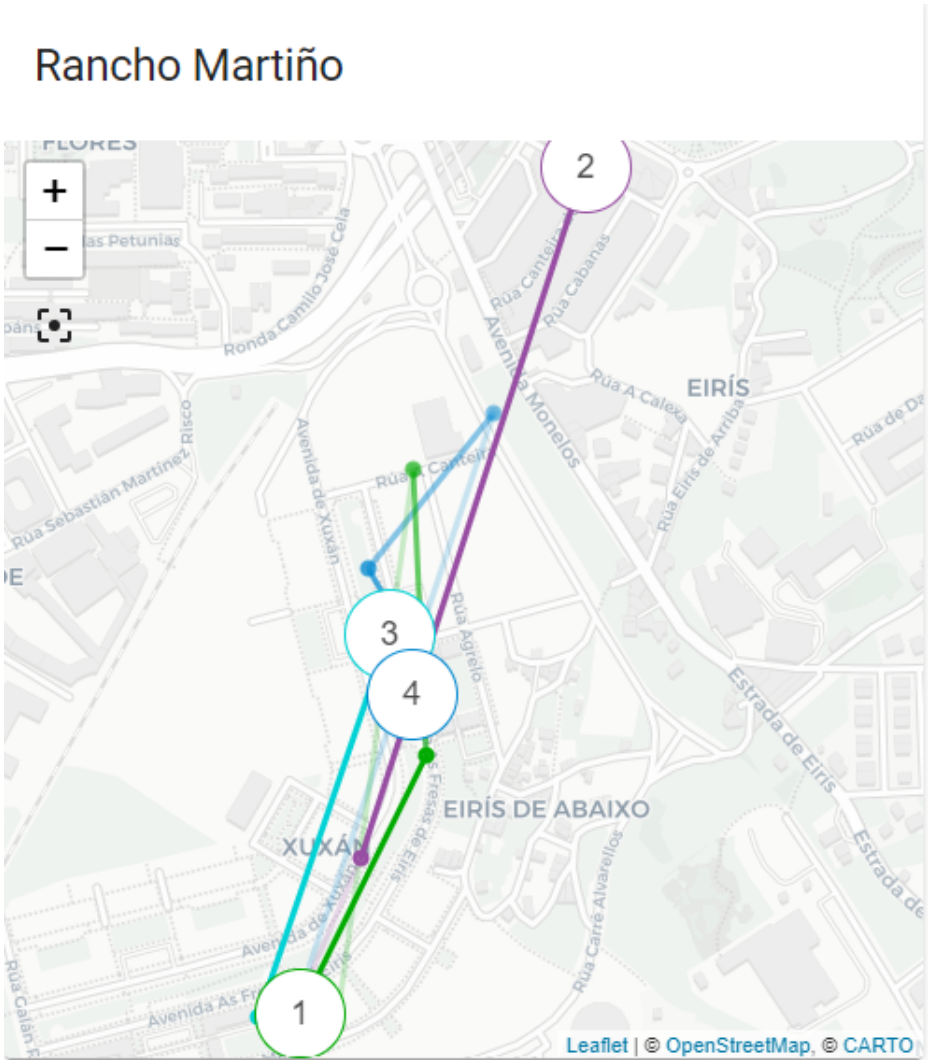


Figura 5.15: Mapa donde se muestra la posición de cada animal y sus trayectorias en la última hora.

Esta prueba ha sido realizada con los distintos localizadores GNSS detallados en la sección 3.2.

Cada uno de los localizadores se había configurado con distintas frecuencias de envío, por eso, en el mismo periodo de tiempo, unos dispositivos tienen más puntos de actualización que otros.

– Las fincas ocupadas las últimas semanas o meses. Con esta carta, mostrada en la figura 5.16, el ganadero podrá planificar las próximas semanas de pastoreo basándose en un historial reciente de la ocupación de las fincas.

Con esta carta, apenas se distinguen los movimientos de cada animal, siendo su principal utilidad saber cuáles son las zonas de las fincas que están más descansadas para planificar las siguientes jornadas de pastoreo.

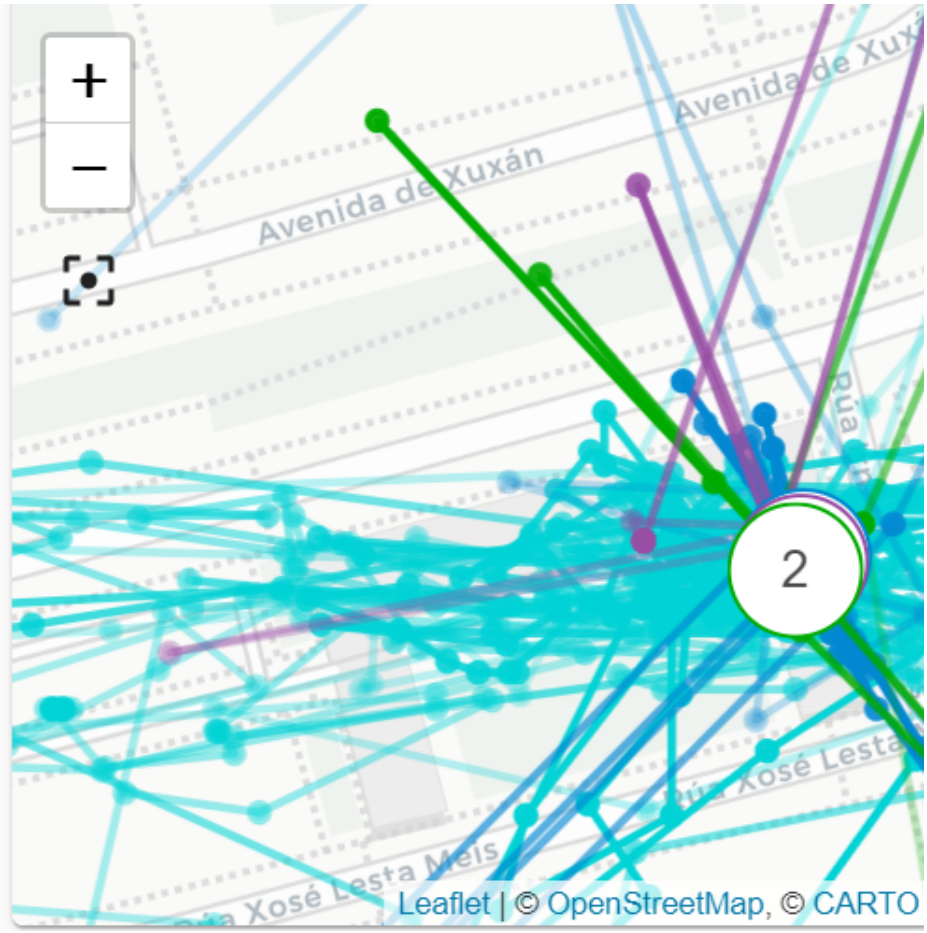


Figura 5.16: Mapa que muestra el historial de los últimos 15 días.

- Una pila vertical donde se muestra información sobre cada tanque. Como se muestra en la figura 5.17, estas cartas indican los litros de agua disponibles en el tanque, con su temperatura y un gráfico que muestra su historial reciente.

Figura 5.17: Carta de monitorización de un tanque de agua móvil de 3 500 litros de capacidad máxima en dos momentos diferentes.

- Botón de control de la valla electrificada. Con este botón se controla el funcionamiento del pastor eléctrico, que da corriente a la valla, logrando apagarla o encenderla. Además, este botón cambiará de color dependiendo del estado de la misma. En la figura 5.18 se muestran los dos posibles estados del botón.

(a) Valla apagada. (b) Valla encendida.

Figura 5.18: Botón de control de la valla electrificada con sus dos posibles estados.

5.6 Gestión de alertas

Cada tipo de alerta irá dirigida a las personas encargadas de su gestión. Estas alertas, deben poder ser enviadas a un destinatario que no tenga instalada la aplicación, ofreciendo así más flexibilidad al ganadero. Por este motivo, se ha decidido que puedan ser enviadas mediante una aplicación de mensajería móvil. Dentro de este tipo de aplicaciones, se ha utilizado la más extendida en el mercado, que es WhatsApp® [75].

Una de las formas de utilizar el servicio de mensajería de WhatsApp® de forma automatizada, es utilizando la API de Callmebot [76], Para ello, se requiere enviar un mensaje de autorización vía WhastsApp, al número +34 644 78 49, desde el dispositivo que se desean recibir las alertas, con el texto ”I allow callmebot to send me messages”. Al cabo de unos minutos, se recibe un mensaje de confirmación con una clave, que será utilizada en el siguiente paso.

Para la integración de Callmebot con Node-RED, acceder a ‘Settings” ‘> “Palette’ y buscar el paquete de nodos que gestiona esta integración, que es ”node-red-contrib-whatsapp-cmb”. Seleccionar el paquete y pulsar “install”.

Una vez instalado el paquete anterior, estará disponible el nodo ”Send Message” en el listado de nodos de Node-RED. Añadir este nodo al editor de flujo y acceder a sus propiedades. Aparece una ventana como la mostrada en la figura 5.19.

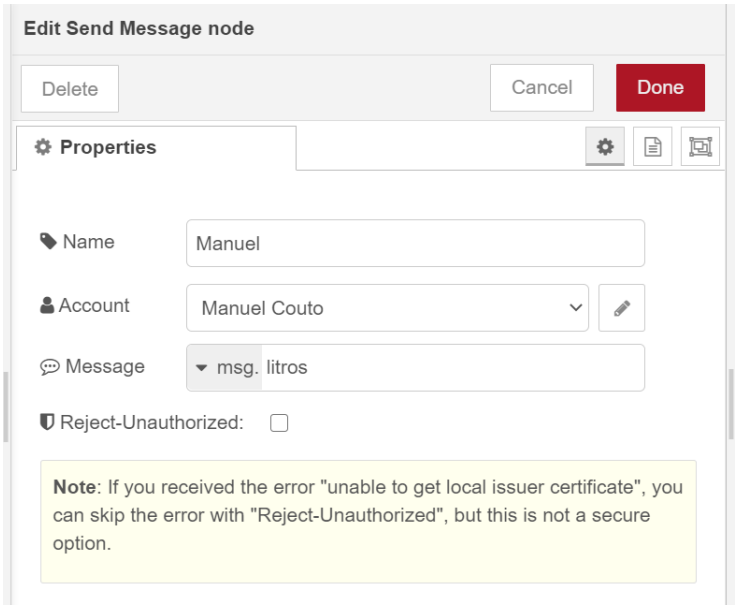


Figura 5.19: Ventana principal de configuración del nodo de WhatsApp® en Node-RED.

Para crear un nuevo usuario de WhatsApp® al que poder notificar, seleccionar "Add new node-red-contrib-whatsapp-cmb-account". Al pulsar en el icono de la derecha, aparece la ventana mostrada en la figura 5.20, en la que se debe de introducir el nombre del destinatario, su número de teléfono e indicará la "API-KEY" recibida por WhatsApp®.

Figura 5.20: Configuración de una cuenta de WhatsApp® en Node-RED.

En el campo "Message", indicar el campo de entrada del nodo que será enviado en la alerta. Para este proyecto, se considera que el ganadero debe recibir un aviso si el contenido del tanque es inferior a 600 litros. En este caso, el ganadero recibirá un mensaje como el mostrado en la figura 5.21.

Figura 5.21: Mensaje de alerta recibido por el ganadero vía WhatsApp® avisando de que el nivel de agua del tanque es bajo.

Otro de los tipos de alerta enviada, al detectar que se ha pulsado el botón de alarma de uno de los dispositivos Dragino [26], o cuando el nivel de batería baja de 2.8 V, voltaje mínimo para

el funcionamiento del localizador GNSS. Se muestra en la figura 5.22, un mensaje disparado por el botón de alarma en el que también se informa del estado de la batería.

Figura 5.22: Mensaje de alarma y de batería baja recibido por el ganadero vía WhatsApp®.

Capítulo 6

Conclusiones y líneas futuras

En este último capítulo de la memoria, se presenta la situación final del trabajo, las lecciones aprendidas y la relación de las competencias de la titulación con la mención de Ingeniería de Computadores. También se incluye un análisis del presupuesto del proyecto, así como las líneas futuras del trabajo realizado.

6.1 Conclusiones

En este TFG se han cubierto las especificaciones y se han alcanzado los objetivos propuestos inicialmente, consiguiendo desarrollar las funcionalidades que, a continuación, se detallan:

- Diseñar y desarrollar la arquitectura de un sistema desasistido, que permita monitorizar y controlar distintos parámetros de una explotación de ganadería extensiva de forma remota.

  Se ha diseñado una arquitectura basada en la nube de una red de comunicaciones abierta, tal y como se muestra en la figura 1.3. Los nodos finales se comunican con el servidor central de Internet of Things (IoT) [66] por medio de las puertas de enlace. Este servidor es el encargado de comunicarse de forma bidireccional con el servidor de aplicaciones alojado en la oficina de la explotación.

- Diseñar y construir un sistema para monitorizar el nivel de llenado y temperatura de un depósito de agua móvil. Además, generar alertas cuando el nivel de agua del tanque sea inferior a una cantidad predefinida.

  Se ha logrado implementar esta funcionalidad instalando un sensor de ultrasonidos en la parte superior del tanque y un sensor de temperatura en contacto con el agua, ambos

conectados a un microcontrolador con conectividad Long Range (LoRa) [50]. El envío de las alertas generadas se ha decidido hacerlo vía WhatsApp®, debido a ser la aplicación de mensajería más extendida [75]. De esta forma, se podrá configurar el sistema, para enviar alertas a distintos individuos sin necesidad de tener instalada la aplicación.

- Diseñar y construir un sistema que permita controlar la valla eléctrica de forma remota.

  Para controlar el pastor eléctrico se utiliza un relé manejado por un microcontrolador con conectividad LoRa. Este relé realiza las funciones de un interruptor, que puede ser accionado desde cualquier sitio con conexión a internet.

- Utilizar distintos dispositivos de localización y transmisión mediante el uso de tecnologías Lower Power Wide Area Network (LPWAN) [9].

  Para lograr llevar a cabo esta funcionalidad, se han utilizado dispositivos de localización Global Navigation Satellite System (GNSS) [77] con pequeñas baterías, para sujetar al cuello de cada animal. Los datos de las posiciones recibidas son transmitidas al servidor de IoT mediante Long Range Wide Area Network (LoRaWAN) [50].

  De los dispositivos utilizados, se ha realizado una comparación y un análisis en la sección 3.2 y se ha seleccionado al localizador Dragino LGT92-AA 3.2.3 [57] como el más adecuado para el proyecto, siendo el de menor consumo y el de mayor precisión.

- Integrar todos los dispositivos hardware y poner en marcha el sistema y sus funcionalidades: plataforma de IoT [66], alertas, panel de control, etc.

  El sistema ha sido desarrollado instalando Home Assistant (HA) [35] como software central del servidor, sobre una Raspberry Pi [31].

  Este planteamiento ha sido de gran ayuda para la integración inicial de los dispositivos y su gestión. En los inicios del proyecto era directamente sobre HA donde se desarrollaba la lógica de negocio de la aplicación, aunque, de forma progresiva, se ha migrado gran parte de esta lógica a Node-RED [36]. Esta herramienta ha facilitado la integración del servidor con distintas APIs, mediante la utilización de nodos disponibles en la herramienta, evitando el desarrollo de código a bajo nivel y permitiendo, a su vez,

implementar nodos propios.

En cuanto al uso de redes LPWAN, existen múltiples comunidades y sitios web especializados, que permiten la búsqueda e intercambio de información, siendo de gran ayuda para aligerar la curva de aprendizaje. La existencia de numerosas comunidades pone de manifiesto el alto grado de madurez de estas tecnologías.

Las pruebas puntuales del sistema se han desarrollado en colaboración con una explotación ganadera establecida en Zas, un pequeño municipio de la provincia de A Coruña. Esta empresa cuenta con alrededor de 100 animales en semilibertad.

En las instalaciones de la ganadería se realizaron pruebas de cobertura, instalando la puerta de enlace en su oficina central. En base a los resultados obtenidos, se observa que estas redes con conectividad LoRa, tienen mayor efectividad en zonas de campo abierto, en comparación con entornos urbanos, consiguiendo establecer comunicaciones a distancias superiores a 10 Km, por lo que se considera una tecnología especialmente acertada para la solución del problema abordado.

Otra de las pruebas de campo realizadas, ha sido con el sistema de monitorización de nivel de agua del tanque, verificando la validez de los resultados obtenidos.

Debido a la reciente implantación de este tipo de tecnologías en el campo de la ganadería y viendo su alto grado de aceptación, se prevé un gran recorrido de estas, no solo en el mundo ganadero, sino en campos como la agricultura y, en general, la monitorización de distintos parámetros en el rural.

6.2 Coste del sistema

De cara a conocer el coste del sistema, se detalla, en diferentes tablas, el gasto realizado en la adquisición del material. La división de las tablas está realizada según las funcionalidades de los elementos hardware del sistema.

- Se muestra en la tabla 6.1 el gasto realizado para la adquisición de elementos comunes.

Concepto	Cantidad	Precio unitario	Importe
Gateway Dragino LPS8	1	87.71 €	87.71 €
Raspberry Pi 4 Model B	1	34.65 €	34.65 €
Tarjeta microSD	1	6.55 €	6.55 €
Cable de red	2	1.75 €	3.50 €
Transformador corriente	2	1.82 €	3.64 €
Total			136.05 €

Tabla 6.1: Presupuesto de los elementos comunes utilizados en el proyecto.

- En la tabla 6.2 se detalla el gasto realizado para la adquisición de dispositivos ubicados en el ganado.

Concepto	Cantidad	Precio unitario	Importe
Dragino LGT92-LI	1	35.12 €	35.12 €
Dragino LGT92-AA + batería	1	34.14 €	34.14 €
HTCC-AB02S + batería	1	26.38 €	34.38 €

Tabla 6.2: Presupuesto del material utilizado el proyecto para la localización de los animales.

- En la tabla 6.3 se presenta el gasto realizado para el montaje de cada nodo final de monitorización de nivel de agua.

Concepto	Cantidad	Precio unitario	Importe
Arduino MKR WAN 1310	1	39.93 €	39.93 €
Módulo JSN-SR04T	1	3.92 €	3.92 €
Sensor DS18B20	1	1.65 €	1.65 €
Cables	1	1.00 €	1.00 €
Caja estanca IP67	1	6.00 €	6.00 €
Batería 2 000 mah	1	4.00 €	4.00 €
Total			56.50 €

Tabla 6.3: Presupuesto del material utilizado para el sistema de monitorización de un tanque de agua.

- Se detalla en la tabla 6.4 el gasto realizado en la adquisición del material para el sistema de control del pastor eléctrico.

Concepto	Cantidad	Precio unitario	Importe
HTCC-AB01	1	22.02 €	22.02 €
Relé KY-019	1	0.84 €	0.84 €
Caja estanca IP67	1	6.00 €	6.00 €
Cables	1	1.00 €	1.00 €
Batería 10 000 mah	1	8.00 €	8.00 €
Total			37.86 €

Tabla 6.4: Presupuesto del material utilizado para el sistema de control del pastor eléctrico.

En este análisis de costos, no está incluido el beneficio, ni se trata de un plan de negocio, por lo que no se computan las horas de dedicación del alumno ni del tutor, debido a que son consideradas tiempo de formación. Tampoco se incluye el gasto de corriente, desplazamientos a la finca para las pruebas, la conexión a internet, ni la amortización del hardware de uso común, como son el ordenador, el router, el teléfono móvil u otros dispositivos de acceso al sistema.

En la figura 6.5, se realiza un presupuesto de material para una ganadería con 100 animales, con características similares a la explotación en la que se realizaron las pruebas. Los precios de los elementos son unitarios. En caso de realizar compras por volumen, los precios tendrían una reducción importante.

Concepto	Cantidad	Precio unitario	Importe
Elementos comunes	1	136.05 €	136.05 €
Monitorización tanque	2	56.50 €	113.00 €
Localización de ganado	100	34.14 €	3 414.00 €
Control pastor eléctrico	1	8.00 €	8.00 €
Total			3 671.05 €

Tabla 6.5: Presupuesto del material utilizado para un proyecto con 100 animales.

Estos gastos sería rápidamente amortizados por la ganadería gracias a todas las ventajas y servicios de valor añadido que le proporcionan. Se debe considerar que el coste de una res puede llegar a los 3 000 €, así que la inversión y la reducción de la probabilidad de su pérdida hace muy atractivo un sistema como el presentado en este proyecto.

6.3 Relación con las competencias de la titulación

El desarrollo de este TFG muestra competencias adquiridas durante el transcurso de la titulación que eran objetivo de la misma, como son:

- Diseñar y programar sistemas empotrados basados en microprocesadores y de comunicaciones.
- Programar, analizar y evaluar sistemas de tiempo real.
- Comprender, aplicar y gestionar la garantía y seguridad de infraestructuras informáticas.
- Diseñar, desplegar, administrar y gestionar infraestructuras y redes de computadores.

6.4 Líneas futuras

El tiempo estipulado para la realización de este proyecto, aunque ha sido suficiente para cumplir los objetivos propuestos inicialmente, no ha permitido desarrollar algunos aspectos, que es interesante incorporar a las líneas futuras, que a continuación se detallan:

- Centralizar la Bases de Datos (BBDD) con el sistema de gestión de bases de datos de series temporales InfluxDB, detallado en la sección 2.3.3 [39] y profesionalizar la interfaz gráfica utilizando Grafana [47], software de visualización comentado en la sección 2.3.4.
- Posibilitar la creación de vallas virtuales. El ganadero podrá, de forma dinámica, crear vallas virtuales y recibir alertas si algún animal las cruza.
- Implementación de un sistema inteligente para la mejora de detección de patologías y detección de animales en celo o próximos al parto.
- Incorporar funcionalidades de protección ante ataques de animales salvajes. El sistema detectará situaciones de peligro y activará mecanismos de protección del ganado.

Por último, dado que el nivel de madurez tecnológica actual de este proyecto corresponde a un Technological Readiness Level (TRL) 6, habiendo sido probados los prototipos en un entorno real y siendo capaces de realizar todas las funciones que se requieren, se ha enviado una comunicación de resultados de investigación a la Oficina de Transferencia de Resultados de Investigación (OTRI) para solicitar apoyo y conseguir madurar esta solución.

Apéndices

Apéndice A

Material adicional

A.1 Fragmento de código de transmisión de datos desde un nodo final

```
void loop()
{
  switch( deviceState )
  {
    case DEVICE_STATE_INIT:
    {
#if(AT_SUPPORT)
      getDevParam();
#endif
      printDevParam();
      LoRaWAN.init(loraWanClass,loraWanRegion);
      deviceState = DEVICE_STATE_JOIN;
      break;
    }
    case DEVICE_STATE_JOIN:
    {
      LoRaWAN.displayJoining();
      LoRaWAN.join();
      break;
    }
    case DEVICE_STATE_SEND:
    {
      LoRaWAN.displaySending();
      prepareTxFrame( appPort );
      LoRaWAN.send();
      deviceState = DEVICE_STATE_CYCLE;
      break;
    }
```

```
29      case DEVICE_STATE_CYCLE:
30      {
31        // Schedule next packet transmission
32        txDutyCycleTime = appTxDutyCycle + randr( 0,
        APP_TX_DUTYCYCLE_RND );
33        LoRaWAN.cycle(txDutyCycleTime);
34        deviceState = DEVICE_STATE_SLEEP;
35        break;
36      }
37      case DEVICE_STATE_SLEEP:
38      {
39        LoRaWAN.displayAck();
40        LoRaWAN.sleep();
41        break;
42      }
43      default:
44      {
45        deviceState = DEVICE_STATE_INIT;
46        break;
47      }
48    }
49  }
```

A.2 Principales características y especificaciones de LoRaWAN

Clases

La especificación LoRaWAN define tres tipos de dispositivos. Todos los dispositivos deben implementar la Clase A, mientras que la Clase B y la Clase C son extensiones de la especificación de los dispositivos de Clase A [78].

Los dispositivos de clase A admiten la comunicación bidireccional entre un dispositivo y una puerta de enlace. Los mensajes de enlace ascendente pueden ser enviados en cualquier momento y, a continuación, el dispositivo abre dos ventanas de recepción, en momentos específicos, después de una transmisión de enlace ascendente. Como se muestra en la figura A.1, si el servidor no responde en ninguna de estas ventanas de recepción (situación 1), la próxima oportunidad será después de la próxima transmisión de enlace ascendente desde el dispositivo. El servidor puede responder a la primera (situación 2) o a la segunda ventana de recepción (situación 3), pero no debe utilizar ambas ventanas.

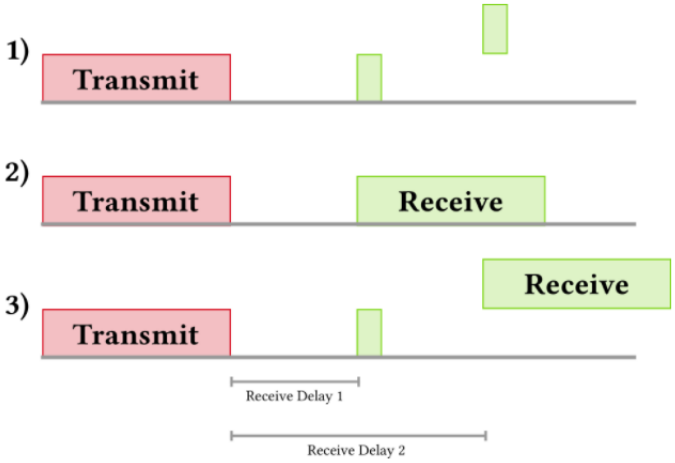


Figura A.1: Esquema de comunicación bidireccional de los dispositivos LoRaWAN Clase A reflejando las tres posibles situaciones de un enlace descendente.

Los dispositivos de Clase B, amplían la Clase A, al agregar ventanas de recepción programadas para los mensajes de enlace descendentes desde el servidor.

Los dispositivos de Clase C, amplían la Clase A, manteniendo abiertas las ventanas de recepción, a menos que estén transmitiendo, como se muestra en la Figura A.2. Esto permite una comunicación de baja latencia, con un alto consumo de energía en relación a los dispositivos de Clase A.

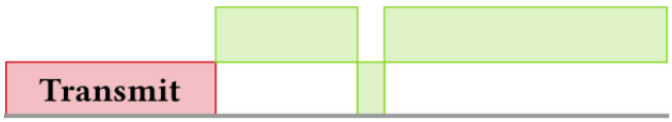


Figura A.2: Esquema de comunicación bidireccional de los dispositivos LoRaWAN Clase C que representa en verde las ventanas de recepción.

Ciclo máximo de trabajo

El ciclo máximo de trabajo de los dispositivos indica la fracción de tiempo que un recurso está ocupado. Se establece en la mayoría de los países en 1% del tiempo.

Seguridad

LoRaWAN especifica una serie de claves de seguridad : NwkSKey, AppSKey y AppKey [79], todas ellas con una longitud de 128 bits. El algoritmo utilizado para esto es AES-128,

similar al algoritmo utilizado en el estándar 802.15.4.

Modos de activación

Para establecer una conexión LoRaWAN se requiere una serie de claves y número de identificación por parte del nodo para lograr el correcto funcionamiento del dispositivo y mantener su seguridad.

Los dispositivos LoRaWAN tienen un identificador único de 64 bits que el fabricante del chip asigna al dispositivo, aunque en ocasiones se puede modificar. Sin embargo, toda la comunicación se realiza con una dirección de dispositivo dinámica de 32 bits de los cuales 7 bits son fijos para TTN, dejando 25 bits que se pueden asignar a dispositivos individuales mediante un procedimiento llamado activación.

Existen dos modos de activación:

- El modo Over The Air Activation (OTAA) sirve para establecer conexiones con mayor seguridad. La sesión se crea cada vez que un dispositivo se conecta a la red, lo que dificulta los robos de sesiones o clonados de dispositivos.
- El modo Activation By Personalisation (ABP) es el modo más sencillo para conectarse, pero presenta algunas desventajas relacionadas con la seguridad.

Lista de acrónimos

ABP Activation By Personalisation. 82

ADR Adaptive Data Rate. 13

AT Attention. 29, 35

BBDD Bases de Datos. 18, 76

BLE Bluetooth Low Energy. 15

BW Bandwidth. 11

CR Coding Rate. 11

DBMS Database Management System. v, 18

DDNS Dynamic Domain Name System. 55

GNSS Global Navigation Satellite System. 14–16, 19, 21, 27–29, 33, 62, 64, 69, 72

GSM Global System for Mobile communications. 19

HA Home Assistant. vi, 17, 19, 48–52, 54–57, 61–63, 72

HACS Home Assistant Community Store. vi, 52–54, 63

HassOS Home Assistant Operating System. 17

ICT Information and Communications Technology. 2

IDE Integrated Development Environment. v, vi, 31–33

IoT Internet of Things. vi, ix, 1, 4–6, 8, 10, 13, 15, 17–21, 23, 24, 34, 35, 37, 71, 72

IP Internet Protocol. 38, 57

ISM Industrial Scientific & Medical. 11

JSON JavaScript Object Notation. 42–44

LoRa Long Range. 10, 11, 15, 16, 21, 24, 30, 31, 72, 73

LoRaWAN Long Range Wide Area Network. v, vii, 11–13, 15, 21, 23, 24, 29, 32, 33, 37, 38, 72, 80–82

LPS Local Positioning System). 16

LPWAN Lower Power Wide Area Network. 9, 10, 72, 73

MAC Media Access Control. 57

MQTT Message Queuing Telemetry Transport. vi, 17, 45, 51, 52, 59–62

NB-IoT Narrowband for the Internet of Things. 19, 20

OAuth Open Authorization. 53

ORM Object Relational Mapper. 19

OTAA Over The Air Activation. 82

OTRI Oficina de Transferencia de Resultados de Investigación. 76

RFID Radio Frequency Identification. 16

SBC Single Board Computer. 16, 17

SF Spreading Factor. 11

SO Sistema Operativo. 17, 49, 50

SQL Structured Query Language. 19

SSH Secure Shell. 51, 52

TFG Trabajo Fin de Grado. 8, 71, 76

TIC Tecnologías de la información y la comunicación. 1, 3

TRL Technological Readiness Level. 76

TSDB Time Series Database. 17, 18

TTN The Things Network. vi, 33, 37, 39–45, 51, 52, 59, 60, 62, 82

VCA Voltaje de Corriente Alterna. 31

VDC Voltaje de Corriente Continua. 31

Bibliografía

[1] "Explicación de ganadería intensiva," 23 de febrero de 2021. [En línea]. Disponible en: https://www.euroinnova.edu.es/blog/que-es-la-ganaderia-intensiva#:~:text=La%20ganader%C3%ADa%20consiste%20en%20la,%2C%20leche%2C%20lana%2C%20etc.

[2] M. Rodríguez and P. Ramil, "Clasificaciones climáticas aplicadas a galicia," Octubre 2007. [En línea]. Disponible en: https://dialnet.unirioja.es/descarga/articulo/3178785.pdf

[3] "Plano general de ordenación municipal de santa comba," Enero 2001. [En línea]. Disponible en: http://www.santacomba.es/wp-content/uploads/2006/11/mapas_planourbanistico.pdf

[4] "Ley 2/2016, de 10 de febrero, del suelo de galicia," Febrero 2021. [En línea]. Disponible en: https://www.boe.es/buscar/act.php?id=BOE-A-2016-3191

[5] "Empresa digitanimal," 23 de febrero de 2021. [En línea]. Disponible en: https://digitanimal.com/ganaderia-extensiva/

[6] R. Liguero, "Una solución basada en nb-iot," Marzo 2018, 23 de febrero de 2021. [En línea]. Disponible en: https://accent-systems.com/es/blog/vacas-conectadas-futuro-ganaderia/

[7] "Coste productos monitorización," 23 de febrero de 2021. [En línea]. Disponible en: https://digitanimal.com/categoria-producto/bovino/

[8] "Coste renovación de servicios," 23 de febrero de 2021. [En línea]. Disponible en: https://digitanimal.com/producto/dispositivo-gps-para-ganado/

[9] "Tecnologías lpwan," 23 de febrero de 2021. [En línea]. Disponible en: https://en.wikipedia.org/wiki/LPWAN

[10] "Lora alliance," 23 de febrero de 2021. [En línea]. Disponible en: https://lora-alliance.org/about-lora-alliance/

[11] "Sigfox," 23 de febrero de 2021. [En línea]. Disponible en: https://es.wikipedia.org/wiki/Sigfox

[12] "Tecnología 4g," 23 de febrero de 2021. [En línea]. Disponible en: https://es.wikipedia.org/wiki/Telefon%C3%ADa_m%C3%B3vil_4G

[13] "Tecnología 5g," 23 de febrero de 2021. [En línea]. Disponible en: https://es.wikipedia.org/wiki/Telefon%C3%ADa_m%C3%B3vil_5G

[14] "Información sobre semtech," 23 de febrero de 2021. [En línea]. Disponible en: https://www.semtech.com/company

[15] "Modulación lora," 23 de febrero de 2021. [En línea]. Disponible en: https://revspace.nl/DecodingLora

[16] "Building a fossasat-1 lora iot ground station," Diciembre 2019, 23 de febrero de 2021. [En línea]. Disponible en: https://www.rtl-sdr.com/building-a-fossasat-1-lora-iot-ground-station/

[17] "Electron rocket," 23 de febrero de 2021. [En línea]. Disponible en: https://es.wikipedia.org/wiki/Electron_(cohete)

[18] "Globos estratosféricos con lora," Febrero 2021, 23 de febrero de 2021. [En línea]. Disponible en: https://oshwdem.org/charlaglobos_estratosfericos/

[19] M. Lima, "Lorawan y el internet de las cosas," Octubre 2019, 23 de febrero de 2021. [En línea]. Disponible en: https://www.larepublica.co/analisis/miguel-allen-lima-2566379/lorawan-y-el-internet-de-las-cosas-2925247

[20] "Antena lorix one," 23 de febrero de 2021. [En línea]. Disponible en: https://www.lorixone.io/en/products

[21] "Soluciones iot arduino," 23 de febrero de 2021. [En línea]. Disponible en: https://www.arduino.cc/pro/hardware/product-family/mkr-family?id=1996559

[22] "Soluciones iot heltec," 23 de febrero de 2021. [En línea]. Disponible en: https://heltec.org/proudct_center/lora/

[23] "Esp32," 23 de febrero de 2021. [En línea]. Disponible en: https://es.wikipedia.org/wiki/ESP32

[24] "Arquitectura arm," 23 de febrero de 2021. [En línea]. Disponible en: https://es.wikipedia.org/wiki/Arquitectura_ARM

[25] "Transceptores semtech," 23 de febrero de 2021. [En línea]. Disponible en: https://www.semtech.com/products/wireless-rf/lora-transceivers

[26] "Productos dragino," 23 de febrero de 2021. [En línea]. Disponible en: https://www.dragino.com/products/products-list.html

[27] "Productos tektelic," 23 de febrero de 2021. [En línea]. Disponible en: https://tektelic.com/catalog

[28] "Sensor de nivel," 23 de febrero de 2021. [En línea]. Disponible en: https://es.omega.com/prodinfo/sondas-de-nivel-medicion.html

[29] "Funcionamiento pastor eléctrico," Febrero 2021. [En línea]. Disponible en: https://www.plumed.es/blog/funciona-pastor-electrico

[30] "Rfid," 23 de febrero de 2021. [En línea]. Disponible en: https://es.wikipedia.org/wiki/RFID

[31] "Raspberry pi 4," 23 de febrero de 2021. [En línea]. Disponible en: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/

[32] "Operating system images," 23 de febrero de 2021. [En línea]. Disponible en: https://www.raspberrypi.org/software/operating-systems/

[33] "Home assistant operating system," 23 de febrero de 2021. [En línea]. Disponible en: https://github.com/home-assistant/operating-system

[34] "The buildroot user manual," 23 de febrero de 2021. [En línea]. Disponible en: https://buildroot.org/downloads/manual/manual.html

[35] "Home assistant," 23 de febrero de 2021. [En línea]. Disponible en: https://en.wikipedia.org/wiki/Home_Assistant

[36] "Node-red," 23 de febrero de 2021. [En línea]. Disponible en: https://nodered.org/

[37] "About the open js fundation," 23 de febrero de 2021. [En línea]. Disponible en: https://openjsf.org/about/

[38] "Dbms popularity broken down by database model," 23 de febrero de 2021. [En línea]. Disponible en: https://db-engines.com/en/ranking_categories

[39] "Influxdb," 23 de febrero de 2021. [En línea]. Disponible en: https://www.influxdata.com/products/influxdb/

[40] "Influxdata," 23 de febrero de 2021. [En línea]. Disponible en: https://www.influxdata.com/solutions/

[41] "Opentsdb," 23 de febrero de 2021. [En línea]. Disponible en: http://opentsdb.net/overview.html

[42] "What is prometheus?" 23 de febrero de 2021. [En línea]. Disponible en: https://prometheus.io/docs/introduction/overview/

[43] "Timescaledb overview," 23 de febrero de 2021. [En línea]. Disponible en: https://docs.timescale.com/latest/introduction

[44] "What is postgresql?" 23 de febrero de 2021. [En línea]. Disponible en: https://www.postgresql.org/about/

[45] "Sqlalchemy," 23 de febrero de 2021. [En línea]. Disponible en: https://www.sqlalchemy.org/

[46] "About sqlite," 23 de febrero de 2021. [En línea]. Disponible en: https://www.sqlite.org/about.html

[47] "Grafana features," 23 de febrero de 2021. [En línea]. Disponible en: https://grafana.com/grafana/

[48] "Objetivos siega," 23 de febrero de 2021. [En línea]. Disponible en: http://siega.complutig.es/proyecto

[49] "Proyecto gelob," 23 de febrero de 2021. [En línea]. Disponible en: https://gelob.es/descripcion-y-objetivos/

[50] "¿qué es lora y lorawan?" 23 de febrero de 2021. [En línea]. Disponible en: https://lorawan.es

[51] "Datasheet jsn-sr04t-2.0," 2019, 23 de febrero de 2021. [En línea]. Disponible en: https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf

[52] "Ds18b20," Diciembre 2020. [En línea]. Disponible en: https://www.jahankitshop.com/getattach.aspx?id=4635&Type=Product

[53] "Módulo arduino mkr wan 1310," Diciembre 2020. [En línea]. Disponible en: https://store.arduino.cc/mkr-wan-1310

[54] "Cubecell gps-6502," 23 de febrero de 2021. [En línea]. Disponible en: https://heltec.org/project/htcc-ab02s/

[55] "Heltec about," 23 de febrero de 2021. [En línea]. Disponible en: https://heltec.org/about_us/

[56] "Air530," Agosto 2020. [En línea]. Disponible en: https://www.mouser.es/new/seeed-studio/seeedstudio-grove-gps-air530-module/

[57] "Lorawan gps tracker," 23 de febrero de 2021. [En línea]. Disponible en: https://www.dragino.com/products/lora-lorawan-end-node/item/142-lgt-92.html

[58] "Stm32l0x2," 23 de febrero de 2021. [En línea]. Disponible en: https://www.st.com/en/microcontrollers-microprocessors/stm32l0x2.html

[59] "Gnss l76-l," 23 de febrero de 2021. [En línea]. Disponible en: https://www.quectel.com/product/l76l.htm

[60] "Cubecell – dev-board," 23 de febrero de 2021. [En línea]. Disponible en: https://heltec.org/project/htcc-ab01/

[61] "Ky-019 5v relais module," Junio 2017. [En línea]. Disponible en: https://datasheetspdf.com/pdf-file/1402030/Joy-IT/KY-019/1

[62] "Ide arduino," 23 de febrero de 2021. [En línea]. Disponible en: https://www.arduino.cc/en/software

[63] "Como transmitir datos en lorawan, reduciendo el payload," Octubre 2018. [En línea]. Disponible en: https://akirasan.net/la-importancia-de-un-buen-payload-en-lorawan/

[64] "Working with bytes," 23 de febrero de 2021. [En línea]. Disponible en: https://www.thethingsnetwork.org/docs/devices/bytes.html

[65] S. Vishal, "Lorawan join procedure," Febrero 2021. [En línea]. Disponible en: https://www.researchgate.net/figure/LoRaWAN-join-procedure_fig2_325575837

[66] "The things network," 23 de febrero de 2021. [En línea]. Disponible en: https://www.thethingsnetwork.org/community

[67] "Nueva cuenta de usuario en the things network," 23 de febrero de 2021. [En línea]. Disponible en: https://account.thethingsnetwork.org/register

[68] "Puerta de enlace dragino lps8," 23 de febrero de 2021. [En línea]. Disponible en: https://www.dragino.com/products/lora-lorawan-gateway/item/148-lps8.html

[69] "Descarga de home assistant," 23 de febrero de 2021. [En línea]. Disponible en: https://github.com/home-assistant/operating-system/releases/download/5.10/hassos_rpi4-64-5.10.img.xz

[70] "Live usb," 23 de febrero de 2021. [En línea]. Disponible en: https://es.wikipedia.org/wiki/Live_USB

[71] "Instalación home assistant community store," Febrero 2021. [En línea]. Disponible en: https://hacs.xyz/docs/installation/installation

[72] "Hassio google drive backup," 23 de febrero de 2021. [En línea]. Disponible en: https://github.com/sabeechen/hassio-google-drive-backup

[73] "Duck dns," 23 de febrero de 2021. [En línea]. Disponible en: https://www.duckdns.org

[74] "Zte products," 23 de febrero de 2021. [En línea]. Disponible en: https://www.zte.com.cn/global/products

[75] "Aplicaciones más descargadas del mundo," Marzo 2020. [En línea]. Disponible en: https://www.infobae.com/america/tecno/2020/03/03/cuales-son-las-10-aplicaciones-mas-descargadas-del-mundo/

[76] "Callmebot," 23 de febrero de 2021. [En línea]. Disponible en: https://www.callmebot.com/

[77] "Sistema global de navegación por satélite," 23 de febrero de 2021. [En línea]. Disponible en: https://es.wikipedia.org/wiki/Sistema_global_de_navegaci%C3%B3n_por_sat%C3%A9lite#:~:text=Un%20sistema%20global%20de%20navegaci%C3%B3n,en%20tierra%2C%20mar%20o%20aire.

[78] "Clases de dispositivos lorawan," 23 de febrero de 2021. [En línea]. Disponible en: https://www.thethingsnetwork.org/docs/lorawan/classes.html

[79] "Seguridad lorawan," 23 de febrero de 2021. [En línea]. Disponible en: https://www.thethingsnetwork.org/docs/lorawan/security.html